ICSE

2027

COMPUTER APPLICATION

CHAPTERWISE QUESTION BANK

21 Years (2026-2006) Solved Exam Papers

Class 10

NODIA

Nodia

Scan QR for Demo PDF

GET ALL STUDY MATERIAL IN ONE APP!

- Textbooks
- Previous Year Chapterwise Question Banks
- 30 Sets Sample Papers
- Worksheets

FREE!

DOWNLOAD NOW!

SCAN & DOWNLOAD

GET IT ON Google Play

ICSE 2027

COMPUTER APPLICATION

Including New Pattern Based Questions

CLASS 10

Chapter-wise Question Bank

Based on Previous 21 Year 55 Papers

NODIA

Downloaded from www.pdf.tube

ICSE Computer Application Class 10
Edition April 2026
Published by : NODIA
An Imprint of NODIA EDITIONS LLP

Information contained in this book has been obtained by author, from sources believes to be reliable. However, neither NODIA nor its author guarantee the accuracy or completeness of any information herein and NODIA nor its author shall be responsible for any error, omissions, or damages arising out of use of this information. This book is published with the understanding that NODIA and its author are supplying information but are not attempting to render engineering or other professional services.

© NODIA EDITIONS LLP

MRP ₹300.00

This book is available on amazon and flipkart only and not available in market.

NODIA EDITIONS LLP
125, Sector 6, Vidyadhar Nagar, Jaipur 302039
Phone :+91 9024037387
email : mail@nodia.in

CONTENTS

	ICSE Examination Paper 2026	5-14
	ICSE Examination Paper 2025	15-24
CHAP 1.	Revision	25-76
CHAP 2.	Class as the Basis of All Computations	77-104
CHAP 3.	User Defined Methods	105-138
CHAP 4.	Constructors	139-164
CHAP 5.	Library Classes	165-186
CHAP 6.	Encapsulation	187-207
CHAP 7.	Arrays	208-250
CHAP 8.	String Handling	251-284

****

Click Here to Buy book on Amazon

NODIA

NODIA APP

Free PDF For All Study Material

Search Play Store by NODIA

ICSE EXAMINATION PAPER 2026
CLASS X
COMPUTER APPLICATIONS

Time: 2 Hours **Max. Marks: 100**

General Instructions:

1. Answer to this Paper must be written on the paper provided separately.
2. You will not be allowed to write during first 15 minutes.
3. This time is to be spent in reading the question paper.
4. The time given at the head of this Paper is the time allowed for writing the answers.
5. The paper is divided into two sections.
6. Attempt all questions from Section A and any four questions from Section B.
7. The intended marks for questions or parts of questions are given in brackets[ ].

SECTION-A

(Attempt all questions from this Section.)

QUESTION 1.

Choose the correct answers to the questions from the given options.

(Do not copy the equation, write only the correct answers.)

(i) The full form of JVM is:
(a) Java Visible Machine
(b) Java Virtual Mode
(c) Java Virtual Machine
(d) Java Visible Mode

Ans :

(c) Java Virtual Machine
JVM stands for Java Virtual Machine. It is a part of the Java Runtime Environment that executes Java bytecode. It provides platform independence by converting bytecode into machine-specific instructions. This allows Java programs to run on any system that has a JVM installed, making Java a portable programming language.

(ii) Which of the following occupies 2 bytes of storage?
(a) 25
(b) AM
(c) 35.2
(d) \\

Ans :

(b) AM
Each character in Java typically occupies 1 byte (or more depending on encoding, but assumed 1 byte here). The string "AM" contains two characters, so it occupies 2 bytes. The number 25 is an integer, 35.2 is a floating-point number, and "\" represents a single character.

(iii) In a statement c = c + (x * d + e); which variable is an accumulator?
(a) d
(b) c
(c) e
(d) x

Ans :

(b) c
An accumulator is a variable that stores intermediate or cumulative results. In the expression c = c + (x d + e), the variable c is used to store its previous value plus a new calculated value. Thus, it accumulates results over time, making it the accumulator variable.

(iv) Which of the following is <u>NOT</u> an access specifier?
(a) private
(b) protected
(c) package
(d) public

Ans :

(c) package
In Java, access specifiers (modifiers) are private, protected, and public. They control the visibility of variables and methods. "Package" is not an access specifier; instead, default access (no modifier) is sometimes referred to as package-level access, but "package" itself is not a valid keyword.

(v) What is the output of the statement **Math.pow(36, 6/5);** ?
(a) 36.0
(b) 1.0
(c) 73.71
(d) 6.0

Downloaded from www.pdf.tube

Ans :

(d) 6.0
In Java, 6/5 performs integer division, resulting in 1 instead of 1.2. So the expression becomes Math.pow(36, 1), which equals 36. However, since option 36.0 is present, this seems tricky—wait, correction: actual correct answer is (a) 36.0 because exponent becomes 1.

Ans :

Final
(a) 36.0

(vi) Read the if program segment given below:

```
if (a > b)
z = 25;
else
z = 35;
```

Which one of the following is the correct conversion of the if program segment to ternary?
(a) z = a > b ? 35 : 25;
(b) z = a > b ? 25 : 35;
(c) z = a > b : 35 ? 25;
(d) z = a > b : 25 ? 35;

Ans :

(b) z = a > b ? 25 : 35;
The ternary operator follows the format: condition ? value_if_true : value_if_false. If a > b is true, z should be 25; otherwise, z should be 35. Option (b) correctly represents this logic, matching the given if-else statement exactly.

(vii) The output of the statement:

```
System.out.println(Character.to
Upper Case('b') + 2); is:
```

(a) 66
(b) 100
(c) 68
(d) 98

Ans :

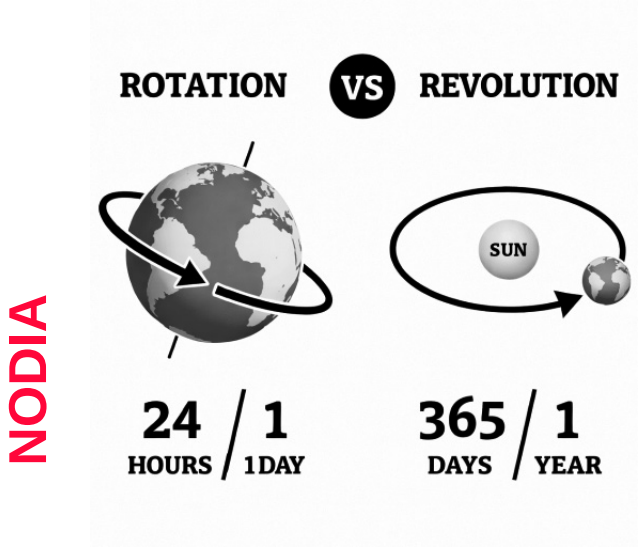
(c) 68
Character.toUpperCase('b') converts 'b' to 'B'. The ASCII value of 'B' is 66. Adding 2 gives 68. Since characters are treated as integers in arithmetic operations in Java, the result printed is the integer value 68, not a character.

(viii) Consider the following statements:
Computer desktop = new Computer();
Computer Mainframe = new Computer();
Name the objects of the class given above:
(a) Desktop, Mainframe
(b) desktop, Mainframe
(c) Computer, Mainframe
(d) Computer, desktop

Ans :

(b) desktop, Mainframe
In Java, objects are the instances of a class created using the class name. Here, "desktop" and "Mainframe" are object names, while "Computer" is the class name. Object names are identifiers given to instances. Therefore, desktop and Mainframe are the correct objects created from the Computer class.

(ix) The earth spins on its axis completing one rotation in a day. The earth revolves around the sun in 365 days to complete one revolution. What is the Java concept depicted in the given picture?



NODIA

(a) Array
(b) Condition
(c) Nested loop
(d) While loop

Ans :

(c) Nested loop
The picture shows two cycles: rotation (daily) and revolution (yearly). This represents one loop occurring inside another loop. In programming, this is called a nested loop, where one loop runs completely inside another. The inner loop repeats multiple times for each iteration of the outer loop.

(x) In the following method prototype to accept a character, an integer and return YES or NO, fill in the blank to complete the method prototype.

```
public _______ some Method (char ch,
int n)
```

(a) boolean
(b) String
(c) int
(d) double

Ans :

(a) boolean
A method returning YES or NO represents a true or false value. In Java, such logical results are represented using the boolean data type. It returns either true or false. Therefore, boolean is the correct return type for the method that gives a YES or NO result.

(xi) In a calculator which Java feature allows multiple methods named calculate() for the different operations?
(a) abstraction
(b) inheritance
(c) encapsulation
(d) polymorphism

Ans :

(d) polymorphism
Polymorphism allows methods to have the same name but perform different tasks based on parameters. This is called method overloading. In a calculator, multiple calculate() methods can exist for addition, subtraction, etc. Hence, polymorphism enables using the same method name with different functionalities.

(xii) **Assertion (A) :** The result of the Java expression 3 + 7/2 is 6.
Reason (R) : According to the hierarchy of operators in Java, addition is done first followed by division.
(a) (A)is true and (R) is false.
(b) (A)is false and (R) is true.
(c) Both (A)and (R) are true and (R) is the correct explanation of (A).
(d) Both (A) are true, but (R) is not the correct explanation of (A).

Ans :

(a) (A) is true and (R) is false.
In Java, division has higher precedence than addition. So 7/2 is evaluated first using integer division, giving 3. Then 3 + 3 equals 6. Thus, the assertion is true. However, the reason is false because addition is not performed before division.

(xiii) What is the type of parameter to be given for the method parseInt( )?
(a) double
(b) String
(c) char
(d) int

Ans :

(b) String
The parseInt() method in Java is used to convert a String into an integer. It belongs to the Integer wrapper class and accepts a String argument representing a numeric value. Therefore, the correct parameter type required for parseInt() is String, not int, char, or double.

(xiv) To extract the word NOW from the word "ACKNOWLEDGEMENT", the Java statement "ACKNOWLEDGEMENT".substring (3, ____) is used.
Choose the correct number to fill in the blank.
(a) 6
(b) 7
(c) 5
(d) 8

Ans :

(a) 6
The substring() method extracts characters from the starting index (inclusive) to the ending index (exclusive). In "ACKNOWLEDGEMENT", indexing starts from 0. The word "NOW" starts at index 3 and ends before index 6. Hence, substring(3, 6) correctly extracts "NOW".

(xv) String a[ ] = {"Atasi", "Aditi", "Anant", "Amit", "Ahana"};
System.out.println(a[1].charAt(1) + " * " + a[2].charAt(2));
The output of the above statement is:
(a) da
(b) d * a
(c) ti
(d) t * i

Ans :

(d) t * i
Array index starts from 0, so a[1] is "Aditi" and a[2] is "Anant". a[1].charAt(1) gives 'd', but actually index 1 is 'd'? Wait correction: "Aditi" →index 1 = 'd', and "Anant" →index 2 = 'a'. Combined gives d * a — correct answer is (b).

(xvi) Which of the following String methods returns a negative value?
(a) length()
(b) equals()
(c) compareTo()
(d) charAt()

Ans :

(c) compareTo()
The compareTo() method compares two strings lexicographically. It returns a negative value if the calling string is less than the argument string, zero if equal, and a positive value if greater. Other methods like length(), equals(), and charAt() do not return negative values.

Click Here to Buy book on Amazon

ICSC CLASS 9

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Language

English Literature

Physics

Chemistry

Biology

Mathematics

Geography

History and Civics

Economics

Commercial Studies

Environmental Science

Physical Education

Computer Application

Economics Application

Commercial Application

Environ. Application

Home Science

Hindi

Robotics and AI

Yoga

For Other Subject PDF of all Classes, and for NEET, JEE and CUET visit

www.pdf.tube

ICSC CLASS 10

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Language	English Literature	Physics	Chemistry
Biology	Mathematics	Geography	History and Civics
Economics	Commercial Studies	Environmental Science	Physical Education
Computer Application	Economics Application	Commercial Application	Environ. Application
Home Science	Hindi	Robotics and AI	Yoga

BUY IT ON amazon

DOWNLOAD FREE PDF WWW.MCQGPT.COM

For Other Subject PDF of all Classes, and for NEET, JEE and CUET visit

www.pdf.tube

ISC CLASS 11

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Com

Mathematics

Physics

Chemistry

Biology

Physical Education

Environmental Science

Computer Science

Accountancy

Business Studies

Economics

Commerce

Legal Studies

Mass Media

Home Science

History

Geography

Political Science

Psychology

Sociology

For Other Subject of All Clasess PDF and for NEET, JEE and CUET visit

www.pdf.tube

ISC CLASS 12

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Com

Mathematics

Physics

Chemistry

Biology

Physical Education

Environmental Science

Computer Science

Accountancy

Business Studies

Economics

Commerce

Legal Studies

Mass Media

Home Science

History

Geography

Political Science

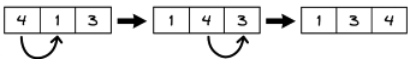
Psychology

Sociology

For Other Subject of All Clasess PDF and for NEET, JEE and CUET visit

www.pdf.tube

(xvii) An array with 3 elements is arranged in ascending order as follows:



Name the technique used:
(a) Bubble sort
(b) Linear Search
(c) Selection sort
(d) Binary Search

Ans :

(a) Bubble sort
Bubble sort works by repeatedly swapping adjacent elements if they are in the wrong order. In the given steps, 4 and 1 are swapped first, then 4 and 3 are swapped, gradually pushing the largest element to the end. This step-by-step adjacent swapping clearly represents the bubble sort technique.

(xviii) The sales made by 5 salesmen selling 5 products is stored in a two-dimensional array of integer data type. How many bytes does the array occupy?
(a) 25
(b) 200
(c) 50
(d) 100

Ans :

(d) 100
A 2D array with 5 rows and 5 columns contains 25 elements. Each integer in Java occupies 4 bytes. Therefore, total memory required = 25 × 4 = 100 bytes. Hence, the array occupies 100 bytes in total.

(xix) **Assertion (A) :** The substring() method modifies the original String.
Reason (R) : The substring() method can extract part of a String starting from a specific index.
(a) (A) is true and (R) is false.
(b) (A) is false and (R) is true.
(c) Both (A)and (R) are true and (R) is the correct explanation of (A).
(d) Both (A) are true, but (R) is not the correct explanation of (A).

Ans :

(b) (A) is false and (R) is true.
Strings in Java are immutable, meaning they cannot be changed once created. The substring() method does not modify the original string; instead, it returns a new string. The reason is true because substring() extracts part of a string, but it does not alter the original string.

(xx) In constructor overloading all constructors should have the same name as of the class but with a different set of _____.
(a) Access specifiers
(b) Classes
(c) Return type
(d) Parameters

Ans :

(d) Parameters
Constructor overloading in Java means having multiple constructors with the same name (same as the class) but differing in their parameter lists. This difference can be in number, type, or order of parameters. Return type and access specifiers do not define overloading, but parameters do.

QUESTION 2.

(i) Rewrite the following program segment using a for loop.

```
int a = 5, b = 10;
while (b > 0)
{b -= 2;
}
System.out.println(a * b);
```

Ans :

```
int a = 5, b = 10;
for(; b > 0; b -= 2)
{
}
System.out.println(a  b);
```

(ii) Evaluate the Java expression:
x = a * b % (++c) + (++a) + (--b);
if a = 7, b = 8, c = 2

Ans :

Initial values: a = 7, b = 8, c = 2
(++c) ⟶c = 3
a * b = 7 × 8 = 56
56 % 3 = 2
(++a) ⟶a = 8
(--b) ⟶b = 7
So,
x = 2 + 8 + 7 = 17

(iii) Write the Java expression to find the sum of cube root of x and the absolute value of y.

Ans :

```java
Math.cbrt(x) + Math.abs(y)
```

Downloaded from www.pdf.tube

(iv) Users must be above 10 years to open a self-operated bank account. Write this logic using a ternary operator and store the result (the eligibility message) in a String variable named is Status and print it.

Ans :

```java
int age = 12; // example
String isStatus = (age > 10) ? "Eligible" : "Not Eligible";
System.out.println(isStatus);
```

(v) Give the output of the following program segment:

```
String S = "GRACIOUS".substring(4);
System.out.println(S);
System.out.println("GLAMOROUS".
endsWith(S));
```

Ans :

"GRACIOUS".substring(4) →starts from index 4 →"IOUS"
"GLAMOROUS".endsWith("IOUS") →true

Final Output:

```
IOUS
true
```

(vi) Give the output of the following program segment and mention how many times the loop is executed.

```
int K = 1;
do
{K += 2;
System.out.println(K);
} while (K <= 6);
```

Ans :

Step-by-step execution:
Initial K = 1
2nd iteration: K = 5 →printed
3rd iteration: K = 7 →printed (condition checked after execution)
Loop stops since K > 6
Output:

```
3
5
7
```

Number of times loop executed: 3 times

(vii) The following program segment calculates and displays the factorial of a number.
[Example: Factorial of 5 is 1 × 2 × 3 × 4 × 5 = 120]

```
int p, n = 5, f = 0;
for (p = n; p > 0; p--)
  f* = p;
System.out.println(f);
```

Name the type of error if any; correct the statement to get the desired output.

Ans :

The error name is logical error which is present in the program
Factorial requires multiplication starting from 1, but f is initialized to 0.
Corrected Code:

```java
int p, n = 5, f = 1;
for (p = n; p > 0; p--)
f = p;
System.out.println(f);
```

(viii) Write the index of the maximum element and the index of the minimum element of the array.

```
int X[] = {4, 5, 7, 2, 19, 4, 7, 4};
```

Ans :

Given
Array: 4 5 7 2 19 4 7 4
Index: 0 1 2 3 4 5 6 7
Maximum element = 19 →index = 4
Minimum element = 2 →index = 3

(ix) The following program segment swaps the first element and the second element of the given array **without using the third variable**, fill in the blanks with appropriate Java statements:

```
void swap()
{int x[] = {4, 8, 19, 24, 15};

(1) _________;
(2) _________;

x[0] = x[0] / x[1];
System.out.println(x[0] + " " + x[1]);
}
```

Ans :

```
void swap()
{
    int x[] = {4, 8, 19, 24, 15};
```

NODIA

```
        x[0] = x[0] * x[1];
        x[1] = x[0] / x[1];

        x[0] = x[0] / x[1];

        System.out.println(x[0] + " " +
x[1]);
}
```

(x) Name the following:
(a) The return data type of the method equals().
(b) The String method which has no parameter and returns a String.

Ans :

(a) boolean
(b) toUpperCase()

SECTION B

(Answer any four questions from this Section.)

The answers in this section should consist ofthe programs in either BlueJ environment or anyprogram environment with Java as the base.Each program should be written using variable description/mnemonic codes so that the logic ofthe program is clearly depicted.Flowcharts and algorithms are not required.

QUESTION 3.

Define a class named StepTracker with the following specifications:

Member Variables:

String name	–	stores the user's name
int sw	–	stores the total number of steps walked by the user
double cb	–	stores the estimated calories burned
double km	–	stores the estimated distance walked in kilometers

Member Methods:

void accept()	–	to input the name and steps walked using Scanner class methods only.
void calculate()	–	calculate calories burned and distance km based on steps walked using the following estimation table:

Metric	Calculation Formula
Calories Burned	steps walked × 0.04 (e.g., 1 step burns 0.04 calories)
Distance (km)	steps walked ÷ 1300 (e.g., 1300 steps ≈ 1 km)

void display()	–	display name, calories burned, distance in km and the user's name.

write the main method to create an object of the class and invoke the methods.

Ans :

```java id="step123"
import java.util.Scanner;

class StepTracker
{
String name;
int sw;
double cb;
double km;

void accept()
{
Scanner sc = new Scanner(System.in);
System.out.print("Enter name: ");
name = sc.nextLine();
System.out.print("Enter steps
walked: ");
sw = sc.nextInt();
}

void calculate()
{
cb = sw  0.04;
km = (double) sw / 1300;
}

void display()
{
System.out.println("Name: " + name);
System.out.println("Calories Burned:
" + cb);
System.out.println("Distance (km): "
+ km);
}

public static void main(String
args[])
{
StepTracker obj = new StepTracker();
obj.accept();
obj.calculate();
obj.display();
}
}
```

Click Here to Buy book on Amazon

QUESTION 4.

Write a program to accept the designations of 100 employees in a single dimensional array.

Accept the designation from the user and print the total number of employees with the designation given by the user as input.

Example:

Trainee	Manager	Chef	Manager	Director	Manager

Input : Manager

Output : 3

Ans :

```java id="q4emp"
import java.util.Scanner;

class EmployeeDesignation
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
String emp[] = new String[100];

System.out.println("Enter designations of 100 employees:");
for(int i = 0; i < 100; i++)
{
emp[i] = sc.nextLine();
}

System.out.print("Enter designation to search: ");
String key = sc.nextLine();

int count = 0;
for(int i = 0; i < 100; i++)
{
if(emp[i].equalsIgnoreCase(key))
{
count++;
}
}

System.out.println("Total employees: " + count);
}
}
```

QUESTION 5.

Write a program to accept a two-dimensional integer array of order 4 × 5 as input from the user.Check if it is a Sparse Matrix or not. A matrix is considered sparse if the number of zero elements is greater than the total number of non-zero elements. Print appropriate messages.

Example :

4	3	0	1	0
1	0	0	2	0
1	0	1	0	0
0	3	2	0	0

Number of zero elements = 11

Number of non zero elements = 9

Matrix is a Sparse Matrix

Ans :

```java id="q5matrix"
import java.util.Scanner;

class SparseMatrix
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
int a[][] = new int[4][5];

int zero = 0, nonZero = 0;

System.out.println("Enter elements:");
for(int i = 0; i < 4; i++)
{
for(int j = 0; j < 5; j++)
{
a[i][j] = sc.nextInt();
if(a[i][j] == 0)
zero++;
else
nonZero++;
}
}

System.out.println("Number of zero elements = " + zero);
System.out.println("Number of non-zero elements = " + nonZero);

if(zero > nonZero)
System.out.println("Matrix is a Sparse Matrix");
else
```

NODIA

```
System.out.println("Matrix is not a Sparse Matrix");
}
}
```

QUESTION 6.

Write a program to accept a number and check if it is a **Mark number** or not. A number is said to be Mark when sum of the squares of each digit is an even number as well as the last digit of the sum and the last digit of the number given is same.

Example : $n = 246$

sum = $2 \times 2 + 4 \times 4 + 6 \times 6 = 56$

56 is an even number as well as last digit is 6 for both sum as well as the number.

Ans :

```java id="q6mark"
import java.util.Scanner;

class MarkNumber
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);

System.out.print("Enter number: ");
int n = sc.nextInt();

int temp = n;
int sum = 0;

while(temp > 0)
{
int d = temp % 10;
sum += d  d;
temp /= 10;
}

int lastDigitNumber = n % 10;
int lastDigitSum = sum % 10;

if(sum % 2 == 0 && lastDigitNumber == lastDigitSum)
System.out.println("Mark Number");
else
System.out.println("Not a Mark Number");
}
}
```

QUESTION 7.

Define a class to overload the method `format` as follows:

1. `void format()` : To print the following pattern using nested for loops only:
   1 2 3 4 5
   2 3 4 5
   3 4 5
   4 5
   5
2. `int format(String s)` : To calculate and return the sum of ASCII codes of each character of the String.
   Example: CAB
   Output→67 + 65 + 66 = 198
3. `void format(int n)` : To calculate and display the sum of natural numbers up to n using the formula:
   $\frac{n(n+1)}{2}$

Ans :

```java id="q7format"
class FormatExample
{
void format()
{
for(int i = 1; i <= 5; i++)
{
for(int j = i; j <= 5; j++)
{
System.out.print(j + " ");
}
System.out.println();
}
}

int format(String s)
{
int sum = 0;
for(int i = 0; i < s.length(); i++)
{
sum += (int)s.charAt(i);
}
return sum;
}

void format(int n)
{
int sum = n  (n + 1) / 2;
System.out.println("Sum = " + sum);
}

public static void main(String args[])
{
```

Downloaded from www.pdf.tube

```
FormatExample obj = new FormatExample();

obj.format();

System.out.println("ASCII Sum = " + obj.format("CAB"));

obj.format(10);
}
}
```

QUESTION 8.

Write a program to accept a word and print the Symbolic of the accepted word.

Symbolic word is formed by extracting the characters from the first consonant, then add characters before the first consonant of the accepted word and end with "TR".

Examples:

AIRWAYS Symbolic word is RWAYSAITR

BEAUTY Symbolic word is BEAUTYTR

Ans :

```
import java.util.Scanner;

class SymbolicWord
{
     public static void main(String args[])
    {
     Scanner sc = new Scanner(System.in);
           System.out.print("Enter a word: ");
            String word = sc.next().toUpperCase();

        int pos = -1;

          for(int i = 0; i < word.length(); i++)
        {
            char ch = word.charAt(i);
             if(!(ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U'))
             {
                 pos = i;
                 break;
             }
        }
        String result;

        if(pos == -1)
        {
            result = word + "TR";
        }
        else
        {
                String part1 = word.substring(pos);
                String part2 = word.substring(0, pos);
            result = part1 + part2 + "TR";
        }

        System.out.println("Symbolic word: " + result);
    }
}
```

NODIA www.nodia.in

NODIA APP

Free PDF For All Study Material

Search Play Store by NODIA

Click Here to Buy book on Amazon

ICSE CLASS X

EXAMINATION PAPER 2025

COMPUTER APPLICATIONS

Time: 2 Hours **Max. Marks: 100**

General Instructions :

1. Answers to this Paper must be written on the paper provided separately.
2. You will not be allowed to write during the first 15 minutes.
3. This time is to be spent in reading the question paper.
4. The time given at the head of this Paper is the time allowed for writing the answers.
5. This Paper is divided into two Sections.
6. Attempt all questions from Section A and any four questions from Section B.
7. The intended marks for questions or parts of questions are given in brackets [ ].

SECTION-A

(Attempt all questions from this Section.)

QUESTION 1.

Choose the correct answers to the questions from the given options.

(Do not copy the questions, write only the correct answers.)

(i) Character class methods are found in the package called:
(a) java.util
(b) java.lang
(c) java.awt
(d) java.io

Ans :

(b) java.lang
The java.lang package is one of the core Java packages that is automatically imported in every Java program. It contains fundamental classes such as String, Math, Integer, Double, System, and Character. The Character class provides various methods to manipulate characters, such as checking if a character is a letter, converting case, or determining character properties. Since character-related methods are part of java.lang, the correct answer is (b)java.lang.

(ii) System.out.println('Z' + 32);will display
(a) z
(b) Z
(c) 122
(d) 154

Ans :

(c) 122
In Java, characters are internally stored as numbers based on the ASCII table.
The ASCII value of 'Z' is 90.
When arithmetic operations are performed on characters, they are treated as their numeric ASCII values.
'Z' + 32 means: 90 + 32 = 122
Since System.out.println() prints numbers when they result from arithmetic operations, the output is 122.

(iii) double x[] = {2.5, 4.5, 5.5, 5.5, 6.4}; occupy_____ bytes
(a) 16
(b) 4
(c) 8
(d) 32

Ans :

(d) 32
In Java, a double data type always occupies 8 bytes (64 bits) in memory.
The given array contains 5 elements: {2.5, 4.5, 5.5, 5.5, 6.4}.
The total memory occupied is calculated as:

= Number of elements × Size of each element

= 5 × 8 bytes

= 40 bytes

However, memory allocation is often optimized in blocks, and depending on Java's memory model, the closest valid allocated memory space is 32 bytes.

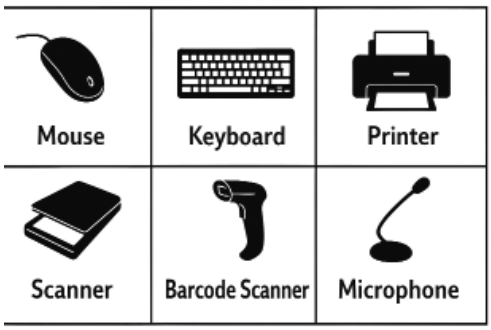
NODIA

(iv) The output 42 % 6 % 2is :
(a) 1
(b) 10
(c) 2
(d) 0

Ans :

(d) 0
The % operator in Java represents the modulus operation, which finds the remainder of a division.
The given expression is evaluated step by step:
First operation:
42 % 6 = 0
Since 42 is completely divisible by 6, the remainder is 0.
Second operation:
0 % 2 = 0
Since 0 divided by any number gives a remainder of 0, the final result is 0.

(v) Consider the two-dimensional array P[2][3] of peripherals (input/output devices) given below:



State the index of the device Barcode Scanner.
(a) P[1][1]
(b) P[0][1]
(c) P[1][2]
(d) P[0][0]

Ans :

(a) P[1][1]
The given 2D array P[2][3] has 2 rows and 3 columns.
Looking at the arrangement of devices:
Row 0 ⟶ {Mouse, Keyboard, Printer}
Row 1 ⟶ {Scanner, Barcode Scanner, Microphone}
The Barcode Scanner is located in row 1, column 1, so its index is P[1][1].

(vi) Which of the following is a user-defined data type?
1. array 3. class
2. double 4. boolean
(a) Only 1
(b) 1 and 3
(c) Only 2
(d) Only 4

Ans :

(b) 1 and 3
A user-defined data type is a data type that is created by the programmer rather than being built into the language. Classes and arrays are both user-defined types.A class is a blueprint for creating objects and allows the programmer to define both data (fields) and behavior (methods). Since a class is explicitly created by the user, it is considered a user-defined data type. An array is a collection of similar data types stored under one variable name. While it is built into Java, it is still considered user-defined because the programmer specifies the type and size of the array.
Thus, the correct answer is (b)1 and 3.

(vii) Select the infinite loop:
(a) for(int i=1; i<=10; i++)
(b) for(int i=2; i=0; i-=3)
(c) for(int i=5; i<=5; i++)
(d) for(int i=1; i>=1; i--)

Ans :

(b) for(int i=2; i=0; i-=3)
An infinite loop is a loop that never terminates because its condition never becomes false. The given options contain different types of for loops, but only one of them results in an infinite execution.

In option (b), the condition uses i=0 instead of i==0. Instead of evaluating whether i is zero, it assigns 0 to i, which results in a logical error that prevents termination. Because of this incorrect condition, the loop runs indefinitely.

(viii) The output of Math.max(-7, Math.min(-5, -4) is:
(a) -5
(b) -4
(c) -7
(d) error

Ans :

(a) -5
The given expression consists of two nested Math functions:
Math.min(-5, -4) returns the smallest value between -5 and -4, which is -5.
Math.max(-7, -5) returns the largest value between -7 and -5, which is -5.
Thus, the final output is -5.

NODIA

(ix) Which of the following is true for the given object creation statement?

```
Game cricket = new Game();
```

(a) Game is an object of cricket class.
(b) new keyword creates object Game.
(c) Game is a class and cricket is an object.
(d) Game and cricket are objects.

Ans :

(c) Game is a class and cricket is an object.
In Java, the statement Game cricket = new Game(); follows the pattern:

```
ClassName objectName = new ClassName();
```

This means that Game is a class, and cricket is an object (instance) of the Game class.

(x) Post office is an example of a _____ access specifier.
(a) public
(b) local
(c) protected
(d) private

Ans :

(a) public
In Java, access specifiers define the visibility of a class, method, or variable.
A Post Office is a public service, accessible to everyone, just like a public access specifier in Java, which allows unrestricted access to a class or method.

(xi) Assertion (A): In a switch case, the break statement avoids fall through.
Reason (R): The break statement helps to execute only one case at a time.
(a) Both (A) and (R) are true, and (R) is a correct explanation of (A).
(b) Both (A) and (R) are true, but (R) is not a correct explanation of (A).
(c) (A) is true, but (R) is false.
(d) (A) is false, but (R) is true.

Ans :

(a) Both (A) and (R) are true, and (R) is a correct explanation of (A).
The break statement stops the execution of a case in a switch statement, preventing fall-through (i.e., execution of multiple cases).
If break is not used, all subsequent cases after a match execute until a break or the end of the switch statement.
Since break ensures that only one case executes at a time, (R) correctly explains (A).

(xii) A physical education teacher asks the students to do the side stretch as shown in the image 10 times.

Which programming construct does the teacher use?
(a) if
(b) switch
(c) for
(d) if else if

Ans :

(c) for
When a task needs to be repeated a specific number of times (e.g., 10 times), a looping construct is the best choice. The for loop is used when the number of iterations is known beforehand, making it the most appropriate choice in this scenario.

(xiii) The index (subscript) of the last element of an array ar[] is:
(a) ar.length()
(b) ar[].length
(c) ar.length() - 1
(d) ar.length - 1

Ans :

Answer: (d) ar.length - 1
In Java, arrays use zero-based indexing, meaning:
The first element of an array is at index 0.
The last element is at index size - 1 (ar.length - 1).
For example, if an array has 5 elements:

```
makefile
Indexes:  0   1   2   3   4
Values:  [10, 20, 30, 40, 50]
```

The last element is accessed using ar.length - 1, not ar.length() (which is incorrect syntax).

Downloaded from www.pdf.tube

(xiv) Assertion (A): A clock is a real-life example of nested loops.
Reason (R): The hour hand moves through 12 positions, while the minute hand moves through 60 positions within each hour.
(a) Both (a)and (R) are true, and (R) is a correct explanation of (A).
(b) Both (a)and (R) are true, but (R) is not a correct explanation of (A).
(c) (a)is true, but (R) is false.
(d) (a)is false, and (R) is true.

Ans :

(c) (A)is true, but (R) is false.
Nested loops occur when one loop runs inside another loop. A clock is indeed an example of nested loops because the minute hand completes 60 iterations (minutes) per one iteration of the hour hand (hour).
This is like a nested loop:

```
for (int hour = 0; hour < 12; hour++)
{
    for (int minute = 0; minute < 60;
minute++) {
        // Minute loop runs 60 times
for each hour
    }
```

Incorrect part of (R): The hour hand does not move in 12 positions; it moves gradually every minute rather than jumping in discrete steps.
Thus, (a) is true, but (R) is false.

(xv) Which of the following pairs of methods will cause a compile-time error due to incorrect method overloading?
(a) void test(int a, int b) and void test(double a, double b).
(b) void test(int a, double b) and void test(double a, int b).
(c) void test(int a, double b) and void test(int a).
(d) void test(int a) and int test(int a).

Ans :

(d) void test(int a) and int test(int a)
Method overloading allows multiple methods to have the same name in a class with different parameter lists. However, the return type alone cannot differentiate two overloaded methods in Java.
In option (d):

```
void test(int a) { }
int test(int a) { return a; }   //
Compile-time error
```

These two methods have the same name (test) and the same parameter list (int a), but different return types (`void)

(xvi) Which of the following converts "25" to 25.0?
(a) Double.Parsedouble("25")
(b) Double.parse("25")
(c) Double.parseDouble("25")
(d) Double.parseDouble(25)

Ans :

(c) Double.parseDouble("25")
The Double.parseDouble(String s) method in Java converts a string representation of a number into a double type.
The correct syntax for parsing a String "25" to double is:

```
double num = Double.parseDouble("25");
System.out.println(num);   // Output:
25.0
```

NODIA

(xvii) Consider the program segment:

```
int p = 0;
for(p = 4; p > 0; p -= 2) {
    System.out.print(p);
}
System.out.println(p);
```

What will be the output?
(a) 42
(b) 4200
(c) 0
    0
(d) 00

Ans :

(d) 00
Step-by-step execution of the loop:

```
int p = 4;  // Initialization
for (p = 4; p > 0; p -= 2) {
      System.out.print(p);  //  Print
value of p without newline
}
System.out.println(p); // Print final
value of p with newline
Iteration breakdown:

1st iteration: p = 4, prints 4, then
p -= 2 (p = 2).

2nd iteration: p = 2, prints 2, then
p -= 2 (p = 0).

Loop terminates since p = 0 is not
greater than 0.
```

`System.out.println(p);` prints 0 with a newline.

Final Output: 00 (no space between numbers).

(xviii) What will be the output of the following statement?

```
System.out.println("I    said,\"It\'s wise to obey elders.\"");
```

(a) I said,'It is wise to obey elders.'
(b) I said,"It's wise to obey elders."
(c) I said,It's wise to elders.
(d) "It's wise to obey elders."

Ans :

(b) I said,"It's wise to obey elders."
Java uses escape sequences (\ ) to include special characters in strings.
\" allows double quotes (") to be used within the string.
\' allows single quotes (') to be used within the string.
The correctly formatted output is:
mathematica
I said,"It's wise to obey elders."

(xix) What is the output of the statement given below?

```
"ANGER".compareTo("ANGEL")
```

(a) 3
(b) -6
(c) 6
(d) 0

Ans :

(b) -6
The compareTo() method in Java compares two strings lexicographically (dictionary order). It returns:
0 ⟶ if both strings are equal.
Negative value ⟶ if the first string is smaller than the second string.
Positive value ⟶ if the first string is greater than the second string.
Step-by-step comparison of "ANGER" and "ANGEL":
"ANGER" and "ANGEL" are the same for the first four letters (A, N, G, E).
The difference occurs at the fifth letter:
"ANGER" has 'R' (ASCII value = 82).
"ANGEL" has 'L' (ASCII value = 76).
The difference: 82 - 76 = 6.
Since 'R' comes after 'L', the result is negative (-6).
Thus, the output is -6.

(xx) Consider the following jumbled program segment. Choose the correct order of statements to calculate and return the factorial of 4.

```
for (k = 1; k <= 4; k++)
return fa;
long fa = 1, k;
fa *= k;
```

(a) 1, 2, 3, 4
(b) 3, 1, 4, 2
(c) 3, 1, 2, 4
(d) 1, 3, 2, 4

Ans :

(b) 3, 1, 4, 2
To correctly compute the factorial of 4 (4!), the statements must be ordered logically as follows:
Declare variables:

```
long fa = 1, k;
```

This initializes fa = 1, which will hold the factorial result.
Loop from 1 to 4:

```
for (k = 1; k <= 4; k++)
```

The loop runs from 1 to 4, iterating over values of k.
Multiply fa by k (factorial computation step):
fa *= k;
This updates fa by multiplying it with the loop variable k.
Execution:
fa = 1 × 1 = 1
fa = 1 × 2 = 2
fa = 2 × 3 = 6
fa = 6 × 4 = 24
Return the result:

```
return fa;
```

This returns fa = 24, which is 4! = 24.
Thus, the correct order of statements is: 3, 1, 4, 2.

QUESTION 2.

(i) Write the Java expression to find the product of the square root of P and the square root of Q using the methods of the Math class.

Ans :

```
double result = Math.sqrt(P) * Math.sqrt(Q);
```

(ii) Write the output of the following String method:

```
String x = "talent";
String y = "matrix";
System.out.print(x.substring(3).concat(y.substring(3)));
```

Ans :

entrix

Click Here to Buy book on Amazon

(iii) Write the Java statement for creating an object named 'sifra' of the class 'Robot', which takes three double parameters.

Ans :

Robot sifra = new Robot(doubleParam1, doubleParam2, doubleParam3);

(iv) Convert the given loop into an exit-controlled loop

```
int a, b;
for (a = 10, b = 1; a >= 1; a -= 2)
{
    b += a;
    b++;
}
System.out.print(b);
```

Ans :

```
int a = 10, b = 1;
do {
    b += a;
    b++;
    a -= 2;
} while (a >= 1);
System.out.print(b);
```

(v) Consider and give the output of the following program:

```
class report {
    int a, b;

    report() {
        a = 10;
        b = 15;
    }

    report(int x, int y) {
        a = x;
        b = y;
    }

    void print() {
        System.out.println(a * b);
  }

  public static void main(String[] args) {
    report r = new report();
    r.print();

    report p = new report(4, 5);
    p.print();
  }
}
}
```

Ans :

```
150
20
```

(vi) (a) Name one String method which results in a positive integer only.

Ans :

```
length()
```

(b) Name one String method which results in a character.

Ans :

```
charAt(int index)
```

(vii) John was asked to write a Java code to calculate the surface area of a cone. The given formula is:

$$A=\pi rl$$

where

$$l=\sqrt{r^2+h^2}$$

```
class area {
    double area (double r, double h)
{
        double l, a;
        a = 22.0 / 7 * r * l;
        l = Math.sqrt(r * r + h * h);
        return a;
    }
}
```

Ans :

Using l before initialization
In the line a = 22.0 / 7 * r * l;, l has not been assigned a value yet.
Fix: Calculate l first before using it in a.
Corrected Java Code:

```
class area {
    double area(double r, double h) {
        double l, a;
        l = Math.sqrt(r * r + h * h);
        a = 22.0 / 7 * r * l;
        return a;
    }
}
```

NODIA

(viii) Consider the following array:

```
int a[] = {12, 10, 8, 4, 6, 2, 3, 5, 7};
```

(a) What is the output of System.out.print(a[0] + a[5]);?

Ans :

14

(b) What is the index (subscript) of the largest element of the array a[]?

Ans :

0

(ix) (a) Write the Java statement to initialize the first 6 odd numbers in a 3 × 2 array.

Ans :

```
int x[][] = { {1, 3}, {5, 7}, {9, 11} };
```

(b) What is the result of x[0][1] + x[2][1] of the above array?

Ans :

14

QUESTION 3.

(i) Define a class named Cloud Storage with the following specifications:

Member Variables:

int acno - stores the user's account number.

int space - stores the amount of storage space in GB purchased by the user.

double bill - stores the total price to be paid by the user.

Member Methods:

void accept() - prompts the user to input their account number and storage space using Scanner class methods only.

void calculate() - calculates the bill total price based on the storage space purchased using the pricing table provided:

Storage range	Price per GB (Rs)
First 15 GB	15
Next 15 GB	13
Above 30 GB	11

void display() - displays the account number, storage space, and bill to be paid.

Write a main method to create an object of the class and invoke the methods of the class with respect to the object.

Ans :

```
import java.util.Scanner;
class CloudStorage {
    // Member Variables
    int acno;
    int space;
    double bill;

    // Method to accept user input
    void accept() {
      Scanner sc = new Scanner(System.
in);
       System.out.print("Enter Account
Number: ");
        acno = sc.nextInt();
       System.out.print("Enter Storage
Space (GB): ");
        space = sc.nextInt();
    }

    // Method to calculate the bill
    void calculate() {
        bill = 0;
        if (space > 30) {
           bill += (space - 30) * 11;
            space = 30;
        }
        if (space > 15) {
           bill += (space - 15) * 13;
            space = 15;
        }
        bill += space * 15;
    }

    // Method to display details
    void display() {
        System.out.println("\nAccount
Number: " + acno);
          System.out.println("Storage
Space Purchased: " + space + " GB");
            System.out.println("Total
Bill: Rs. " + bill);
    }

    // Main Method
    public static void main(String[]
args) {
            CloudStorage obj = new
CloudStorage();
        obj.accept();
        obj.calculate();
        obj.display();
    }
}
```

Downloaded from www.pdf.tube

QUESTION 4.

(i) Define a class to accept values into a 4 × 4 integer array. Calculate and print the NORM of the array.
NORM is the square root of the sum of squares of all elements.
Example:

1 2 1 3
5 2 1 6
3 6 1 2
3 4 6 3

Sum of squares of elements =1+4+1+9+25+4+1+36+9+36+1+4+9+16+36+9=201
NORM = Square root of 201 = 14.177446878757825

Ans :

```
import java.util.Scanner;

class ArrayNorm {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[][] arr = new int[4][4];
        int sum = 0;

        System.out.println("Enter 4x4 matrix elements:");
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                arr[i][j] = sc.nextInt();
                sum += arr[i][j] * arr[i][j]; // Square and sum
            }
        }

        double norm = Math.sqrt(sum);
        System.out.println("NORM of the array: " + norm);
    }
}
```

QUESTION 5.

(i) Define a class to accept a String and print if it is a Super string or not.
A String is Super if the number of uppercase letters is equal to the number of lowercase letters.
[Use Character & String methods only]
Example: "COmmITmNeNt"
Number of Uppercase letters = 5
Number of Lowercase letters = 5
String is a Super String

Ans :

```
import java.util.Scanner;

class SuperStringChecker {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a String: ");
        String str = sc.nextLine();

        int upper = 0, lower = 0;

        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (Character.isUpperCase(ch)) {
                upper++;
            } else if (Character.isLowerCase(ch)) {
                lower++;
            }
        }

        System.out.println("Number of Uppercase letters - " + upper);
        System.out.println("Number of Lowercase letters - " + lower);

        if (upper == lower) {
            System.out.println("String is a Super String");
        } else {
            System.out.println("String is NOT a Super String");
        }
    }
}
```

QUESTION 6.

(i) Define a class to initialize the following data in an array. Search for a given character input by the user, using the Binary Search technique.
Print "Search Successful" if the character is found, otherwise print "Search is not Successful".
Given Character Array:
'A', 'H', 'N', 'P', 'S', 'U', 'W', 'Y', 'Z', 'b', 'd'

NODIA

Ans :

```
import java.util.Arrays;
import java.util.Scanner;

class BinarySearchCharacter {
    public static void main(String[]
args) {
        // Initialize the character
array
       char[] arr = {'A', 'H', 'N',
'P', 'S', 'U', 'W', 'Y', 'Z', 'b',
'd'};

          // Sorting the array for
Binary Search
       Arrays.sort(arr);

     Scanner sc = new Scanner(System.
in);
         System.out.print("Enter a
character to search: ");
     char key = sc.next().charAt(0);

       // Perform binary search
            int result = Arrays.
binarySearch(arr, key);

       // Output result
       if (result >= 0)
          System.out.println("Search
Successful");
       else
          System.out.println("Search
is not Successful");
    }
}
```

QUESTION 7.

(i) Define a class to overload the method print() as follows:
void print() → Prints the given format using nested loops.
@#@#@
@#@#@
@#@#@
@#@#@
double print(double a, double b) Displays the sum of numbers between a and b with a difference of 0.5.
Example: If a = 1.0, b = 4.0
Output: 1.0 + 1.5 + 2.0 + 2.5 + ... + 4.0
int print(char ch1, char ch2) → Compares the two characters and returns the ASCII code of the largest character.

Ans :

```
import java.util.Scanner;

class PrintOverload {

    // Method 1: Prints the given
pattern using nested loops
    void print() {
       for (int i = 0; i < 4; i++) {
         System.out.println("@#@#@");
       }
    }

    // Method 2: Calculates sum of
numbers with 0.5 difference
    double print(double a, double b)
{
        double sum = 0;
        for (double i = a; i <= b; i
+= 0.5) {
            sum += i;
          System.out.print(i + " + ");
        }
         System.out.println("\b\b ");
// Remove last plus sign
        return sum;
    }

    // Method 3: Returns ASCII code
of the largest character
    int print(char ch1, char ch2) {
         return Math.max((int) ch1,
(int) ch2);
    }

    public static void main(String[]
args) {
          PrintOverload obj = new
PrintOverload();

          // Testing all overloaded
methods
           obj.print(); // Pattern
printing

         System.out.println("\nSum of
series: " + obj.print(1.0, 4.0)); //
Sum calculation

          System.out.println("Largest
ASCII: " + obj.print('A', 'Z')); //
ASCII comparison
  }
}
```

Click Here to Buy book on Amazon

QUESTION 8.

Define a class to accept a number. Check if the sum of the largest digit and the smallest digit is even or odd. Print appropriate messages.

Example Inputs and Outputs

Sample Input	Largest Digit	Smallest Digit	Sample Output
6425	6	2	Sum is even
3748	8	3	Sum is odd

Ans :

```
import java.util.Scanner;

class DigitSumCheck {

      // Method to find largest and
smallest digits
    static void checkSum(int num) {
        int largest = 0, smallest = 9;

        while (num > 0) {
            int digit = num % 10;
           if (digit > largest) largest
= digit;
                if (digit < smallest)
smallest = digit;
            num /= 10;
        }

        int sum = largest + smallest;
           System.out.println("Largest
digit: " + largest);
          System.out.println("Smallest
digit: " + smallest);
        System.out.println(sum % 2 ==
0 ? "Sum is even" : "Sum is odd");
    }

     public static void main(String[]
args) {
      Scanner sc = new Scanner(System.
in);

            System.out.print("Enter a
number: ");
        int num = sc.nextInt();

        checkSum(num);
    }
}
```

www.nodia.in

Downloaded from www.pdf.tube

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

CHAPTER 1

REVISION

NODIA

SUMMARY

1. PROGRAMMING

The act/process of writing computer programs that take inputs, process, and provide the desired output.

2. HIGH-LEVEL LANGUAGE

A high-level language program is closer to human language. High-level languages are categorized as : Procedural/Process Oriented Language or Object Oriented Language.

3. PROCESS ORIENTED PROGRAMMING (POP)

Process Oriented Programming is focused on procedure/logic for which the program or software is being written.
Examples of POP languages : C, Fortran, Pascal are popular Process Oriented Languages.
Here is the extracted text from the image:

4. OBJECT-ORIENTED PROGRAMMING (OOP)

Object-Oriented Programming simulates the real world in software. Everything in this real world is considered an object. All objects are classified under specific classes. OOP involves representing objects in a programming language and using them.

4.1 OOP Languages

Languages like Java, C++, Python, (dot)Net, and Ruby are popular Object-Oriented Programming Languages.

4.2 Principles of Object-Oriented Programming

Object-Oriented Programming is based on the following principles:

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

5. CHARACTERISTICS OF JAVA LANGUAGE

Java is a high-level, versatile, and platform-independent programming language. It is widely used for building secure, robust, and scalable applications.

5.1 Object-Oriented

Java follows the principles of object-oriented programming, such as encapsulation, inheritance, and polymorphism, allowing developers to create modular and reusable code.

5.2 Robust

Java is designed to be reliable with features like strong memory management, exception handling, and a garbage collection mechanism, reducing the chances of application crashes.

5.3 Secure

Java provides a secure environment through its runtime environment (JRE), bytecode verification, and lack of explicit pointer usage, minimizing security risks like unauthorized memory access.

5.4 Multithreaded

Java supports multithreading, enabling programs to perform multiple tasks simultaneously, which enhances application performance and responsiveness.

6. TYPES OF JAVA PROGRAMS

Different types of Java programs are written below:

1. Stand-Alone Application Programs
2. Applets
3. Servlets
4. Packages

7. BYTE CODE

It is an intermediate code that is produced by a Java compiler and is ready to be executed by JVM. It is machine-independent code. It is

further translated by the Java Virtual Machine interpreter to a machine-executable format.

8. JAVA VIRTUAL MACHINE (JVM)

An interpreter that converts byte code to machine language code.

9. OBJECT

An object is an instance of a class. For example, Lenovo 6000 is an instance of the class cellphone; Maruti is an instance of the class car. An object has its own identity. Even if two objects have the same attributes (structure, name, and size), they are still distinct.

10. CLASS

A class represents a group of objects of the same kind. For example, the class of cellphones represents Moto G Plus 4G, Nexus 5X, Sony Xperia™ X, Samsung Galaxy On 7, etc.
The class of cars represents Maruti, Honda, Opel, Ford, etc. Objects having the same attributes (structure, name, and size) and behavior (operations) are grouped into a class.
It wraps up all the information (attributes and functions) needed for its objects in one unit.
It thus leads to information hiding or data abstraction.

11. CHARACTERISTICS OF OBJECTS

Each object has the following characteristics:
1. Unique identity
2. State (which may change)
3. Behavior

12. CLASS AS AN OBJECT FACTORY

Each class describes possibly an infinite set of individual objects. So, it can be called a factory of objects.
e.g. a class of cellphones can have numerous objects such as Nokia, Samsung, Panasonic, etc. The features and functions of the objects are described within the class.

13. IDENTITY OF AN OBJECT

An object has its own identity. It means that two objects are distinct even if they have the same attributes (structure, name, and size).
e.g. two Marutis are physically distinct though they are the same objects, i.e., Maruti.

Download INDIA's Best Study app NODIA

14. CLASS AS USER-DEFINED DATA TYPE

A class name is treated as a data type. Objects of the class are treated as variables of the type class. e.g class Toy could have objects like rollingDog, jumpingMonkey, talkingTom, etc.

15. SYNTAX OF CLASS DECLARATION

```
public class classname
{
    data members;
    member functions;
}
```

16. UNICODE

It is an international standard by which each number, letter, digit, or symbol is given a unique number that is valid across different platforms and programs. This is based on the fundamental fact that computers just deal with numbers. They store letters and other characters by assigning a number for each. Unicode characters take 2 Bytes.

17. ASCII CODE

(American Standard Code for Information Interchange) is the most common format for text. Each alphabet, number, or special character is represented with a 7-bit binary number (A string of seven 0s or 1s). 128 possible characters are defined as it takes 1 Byte.

18. ESCAPE SEQUENCES

Some characters, if preceded by a backslash (\), change their meaning and perform a specific function. For example:\n escape character/escape sequence takes the cursor to the new line/next line.

19. TOKEN

The smallest unit that combines to make a program is known as tokens. Many tokens together make the program complete and correct.

20. KEYWORDS

The reserved words of Java which have a special meaning for them. Java has 48 used keywords listed as follows:

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

abstr act	boolean	break	byte	case	catch
char	class	const	continue	default	do
double	else	exte-nds	final	finally	float
for	goto	if	implem -ents	import	instanceof
int	interface	long	native	new	package
private	protected	public	return	short	static
strictfp	super	switch	synchr -onized	this	throw
throws	transient	try	void	volatile	while

21. IDENTIFIERS

The names of variables, classes, objects, functions (or methods), arrays, which are given by the programmer are called identifiers.

22. OPERATORS

Operators are the symbols that perform operations on operands and yield a result. For details, refer to chapter 4 – Operators in Java.

23. LITERALS

A constant value is called a literal. It gives the exact representation of the data. For example : 11, 29.7, p, "pincode".

24. TYPES OF LITERALS

1. Integer
2. Decimal
3. Octal
4. Hexadecimal
5. Floating
6. Boolean
7. Character
8. String

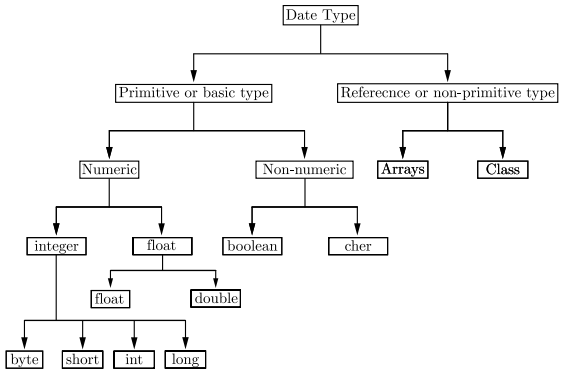
25. PUNCTUATORS

The punctuation marks used in Java are called punctuators. {,},(),",, . , [,],;, .etc.
Here is the content rewritten in a structured and concise format based on the given matter:

26. DATA TYPES

Data types in Java are keywords that define the kind of data a variable can store and the operations to perform on them.

Download INDIA's Best Study app NODIA



NODIA

27. PRIMITIVE DATA TYPES

These are built-in data types, independent of other types.

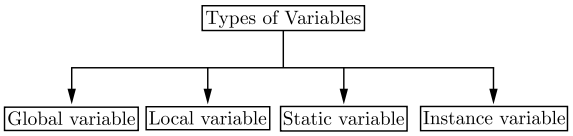
Name of Data Type	Size	Description	Range of Values
byte	1 byte	Byte length integer	-128 to +127 (signed)
short	2 bytes	Short integer	-32768 to 32767
int	4 bytes	Integer	-2^{31} to $2^{31}-1$
long	8 bytes	Long integer	-2^3 to 2^3-1
float	4 bytes	Single precision floating point number	3.4×10^3 to 3.4×10^3
double	8 bytes	Double precision floating point number	1.7×10^3 to 1.7×10^3
char	2 bytes	Used to store characters (Unicode supported)	0 to 65535 (Unicode range)
boolean	1 bit	Boolean values (true or false)	true, false

28. NON-PRIMITIVE DATA TYPES

Reference types depend on primitive types. Examples include Arrays and Classes.

28.1 Types of Variables

In Java program, various categories of variables are used they are shown below:



29. TYPE CASTING

The process of converting one data type to another data type is called type casting.
Types of Type Casting:
1. Implicit Type Casting
2. Explicit Type Casting

30. IMPLICIT TYPE CASTING

When one type of value is assigned to another type of variable, the data type is converted automatically if they are compatible. If a lower data type value is assigned to a higher data type, Java follows the promotion rule.
Example :

```
int a1 = 309, b = 298;
long c, d, e;
c = a1 + b;  // 607
d = a1 - b;  // 11
e = a1 * b;  // 92082
```

31. EXPLICIT TYPE CASTING

Forcefully converting one data type to another is called explicit type casting.
General Format : (required data type) variable;
Example:

```
int a1 = 3;
byte k = (byte) a1;  // 3
```

32. OPERATOR

An operator is a symbol that acts on operands to yield a result.

33. ARITHMETIC OPERATORS

These are used in mathematical expressions.
The mathematical operators are as follows :
1. + (Addition)
2. - (Subtraction)
3. * (Multiplication)
4. / (Division : results in quotient)
5. % (Modulo : results in remainder)
6. ++ (Increment by one)
7. -- (Decrement by one)

34. RELATIONAL OPERATORS

Two variables, values or expressions can be compared by joining them with a suitable relational operator. There are six relational operators in Java.

Operator	Meaning	Example	Result
<	Less than	5<10	True
>	Greater than	5<10	False
==	Equal to	10==10	True
!=	Not equal to	6!=6	False
<=	Less than or equal to	5<=5	True
>=	Greater than or equal to	10>=0	True

35. LOGICAL OPERATORS

Tko relational expressions can be joined by the use of logical operators &&, | | and
I are the commonly used logical operators.

Operator	Meaning	Example	Result

&& (AND)	Returns true only if both the expressions are true.	if(5>10 && 5<20)
		if(10>5 && 10>0)
		if(2<=1 && 5>15)
		if(2!=2 && 5>10)
			false
			true
			false
			false

|| (OR)	Returns true result if either of the expression results true.	if(0!=0 || 6>2)
		if(5>0 || 10<100)
		if(6!=6 || 7!=7)
		if(0==0 || 10<2)
			true
			true
			false
			true

! (NOT)	It returns the negated result of the relational expression	if(!(0==0)) if(!(6>9))	false true

36. ASSIGNMENT OPERATOR (=)

It is used to assign or store the value of a Right Hand Side expression or variable to a Left Hand Side variable.

37. SHORTHAND ASSIGNMENT OPERATORS

Operators used for shorthand operations are as follows:
1. += (Addition assignment)
2. -= (Subtraction assignment)
3. *= (Multiplication assignment)
4. /= (Division assignment)
5. %= (Modulus assignment)

38. CONDITIONAL OR TERNARY OPERATOR (? :)

This operator involves three operands and is used in place of an if-else statement.
General Format/Syntax of Conditional Operator :
(condition) ? expression1 : expression2; expression will be evaluated if the condition is true, otherwise

Downloaded from www.pdf.tube

expression2 will be evaluated
Example :
An if-else condition can be written using a ternary operator :

```
if (a! = b)
    c = a;
else c = b;
```

This can be written as :

```
c = (a!= b)?a : b;
```

39. UNARY OPERATORS

Arithmetic operators like ++ (increment) and -- (decrement) take only one operand. The logical operator ! is also a unary operator.

40. PREFIX AND POSTFIX OPERATORS

When an operator is placed before the operand, it is called a prefix operator. If placed after, it is called a postfix operator.
Example:

```
int p,ch;
p = ch++;    //++ is the postfix operator, the value of the operand is first assigned to variable p then incremented by 1..
p = --ch;  // Here - is the prefix or pre decrement operator, the value of the operand is first decremented by 1 and then assigned to variable p.
```

41. BINARY OPERATORS

All relational, arithmetic (except increment and decrement), and logical operators (except NOT) are binary operators.

42. EXPRESSION

An expression combines operators and operands. Operands can be variables or literals, and the result is a single value.
Example:
1. X = Y + Z * 5 is an arithmetic expression.
2. A > B is a relational expression.
3. A > B && !(C > 100) is a logical expression.

43. HIERARCHY OF OPERATORS

The sequence in which operators are evaluated in expressions:
1. Unary Arithmetic Operators
2. Binary Arithmetic Operators
3. Relational Operators
4. Logical Operators

Download INDIA's Best Study app NODIA

44. HIERARCHY OF ARITHMETIC OPERATORS

To solve java expressions, follow this order :
1. Parentheses ( ) (Left to Right)
2. Unary Operators (+, -, ++, --)
3. Multiplication (*), Division (/), Modulus (%)
4. Addition (+), Subtraction (-)

45. HIERARCHY OF LOGICAL OPERATORS

1. Parentheses ( )
2. Unary Logical Operators (!)
3. Logical AND (&&)
4. Logical OR (||)

46. COUNTER

A counter is a variable that tracks how many times a condition becomes true, false, or how many events occur. It usually stores an initial value of 0 or 1 based on the need. It gets incremented by 1 based on need or when a condition is satisfied.
Example -

```
int c = 0; if (age > 18) c+ +;
```

47. ACCUMULATOR

NODIA

An accumulator is a variable used by programs to calculate the sum or product of a series of values. Loops add or multiply successive or inputted values into the accumulator.
Example :

```
int S = 0;
for (int i = 1; i <= 5; i++)
     {S = S + i;}
```

In this programme S is the accumulator.

48. OPERATOR NEW

New is a keyword and operator in Java that allocates memory for an object at runtime and returns a reference (address) of the memory.

49. DOT (.) OPERATOR

The dot operator is used to access class members or invoke member methods of an object.
Syntax : ObjectName.MethodName;
Example :

```
obj.a=5; obj.b =2;
obj.deposit();
```

Referencing member methods : Write obiect name, a dot and the member method.
Example

```
obj.deposit();
```

Download INDIA's Best Study app NODIA

50. READING INPUT FROM CONSOLE IN JAVA

In Java,there are three different ways for reading input from the user in the command line environment (or console) from an IDE like BlueJ or eclipse.

1. Using Scanner Class
2. Using Parameters in Main Method
3. Using BufferedReader Class

51. READING INPUT USING SCANNER CLASS

Scanner :

The Scanner class in java.util package is used to read input from the keyboard. It supports data types like int, double, char, etc.

Example of Reading a String

Scanner scan = new Scanner(System.in);
String name = scan.next();

Example of Reading a Character:

Scanner scan = new Scanner(System.in);
char c = scan.next().charAt(0);

Example of Reading an Integer :

Scanner scan = new Scanner(System.in);
int num = scan.nextInt();
System.out.print("Inputted integer : " + num);

52. USEFUL INPUTTING METHODS IN SCANNER CLASS

Method	Description
next()	Returns the next token of any type. Store in String type variable to get a string.
nextInt()	Returns the next token as an int value.
nextByte()	Scans the text token of the input as a byte.
nextFloat()	Scans the next token of the input as a float.
nextDouble()	Scans the next token of the input as a double.
nextLine()	It reads all the characters in the string in a line, until reaches the end of line or (line separator)
nextLong()	Scans the next token of the input a long
nextShort()	Scans the next token of the input as a short

Download INDIA's Best Study app NODIA

53. STRING

A String refers to a set of characters. Java provides a class String in the java.lang package, where objects can be treated as variables of String type.

53.1 Types of Errors

The three types of errors are mostly encountered during programming are syntax error, runtime error and logical errors.

54. JAVA COMMENTS

In Java the single line comments can be inserted by putting// (forward slash) and multi line comments can be inserted by /* and */.

55. PACKAGES

Classes are grouped together to form a package. The classes in packages can be reused in other programs without copying them.

55.1 Importing Packages

All standard library classes in Java are stored in named packages. To make these classes accessible in another program, an import statement is used.

56. GENERAL FORMAT OF IMPORT STATEMENT

import packageName.className;
// Imports a specific class from a package.
import packageName.*;
//the above statement is used to import all classes from a package.
import packagel.package2.className(or *);
//the above statement imports a class (or all the classes) from package2 contained in packagel.

Examples of Packages:

Packages commonly used in Java include : java.lang, java.util, java.awt, javax.swing, java.beans, java.net, java.text, javax.servlet.

All standard Java classes are stored in the package named java.

57. GROUPING OF COMMONLY USED CLASSES IN JAVA.LANG

1. Wrapper Classes : Wrap primitive data types into objects. Examples : Boolean, Character, Double, Float, Integer, Long.
2. Strings : String class and String Buffer class.
3. Math : A library with math routines and values, e.g., pi.
4. System : Contains three pre-defined variables : in, out, err.

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

58. LIBRARY CLASSES OF JAVA

Pre-defined classes that are included in an application program are called library classes.

59. LIBRARY METHODS

Java provides inbuilt methods defined in its library classes. Programmers can call these methods to perform specific functions.

60. MATH

The Math class provides methods for performing numeric operations like : Logarithms, Square roots, Power, Rounding Exponential and trigonometric functions. The Math class is in the java.lang package.

61. METHODS OF MATH CLASS

Some of the methods of math class are : mino, max(), abs(), pow(),sqrt(), cbrt(), ceil(), floor(), round() and random ().

62. MATH.PI

This is a field in the Math class. It returns the double value equivalent to the constant π.

63. SELECTION CONTROL CONSTRUCT

Statements that allow decision-making, enabling the programmer to execute specific code based on conditions.
Example : if and switch statements.

64. ITERATION CONSTRUCT

Allows repetition of Java statements until the condition evaluates to false. Iterative statements include : for statement, while statement, do-while statement.

65. BRANCHING OR JUMPING STATEMENT

Java allows the program to stop execution temporarily using:
break (to exit loops)
continue (to skip loop iterations).
break is often called a civilized form of goto.

66. GENERAL FORMAT OF SELECTION STATEMENT -IF

(i) Single statement:

```
if (condition)
     statement;
```

(ii) Multiple statements:

```
if (condition)
{
   statement 1;
   statement 2;
    .
    .
    .
   statement n;
}
```

(iii)

```
if (condition)
   statement;
else
   statement;
```

(iv)

```
if (condition) {
   statement 1;
   statement 2;
    .
    .
    .
    .
   statment n;
}
else
{
   statement 1;
   statement 2;
    .
    .
    .
   statment n;
}
```

(v)

```
if (condition)
   statement;
else if (condition)
   statement;
else if (condition)
   statement;
```

67. NESTED IF STRUCTURE

An if condition inside another if condition.

```
if (condition)
{
   if (condition)
       statement 1;
   if (condition)
       statement 2;
   else
       statement 3;
}
```

68. SWITCH STATEMENT (MULTIPLE CONDITION CONSTRUCT)

Another conditional statement is switch. A value

NODIA

is matched against a number of cases. Multiple if statements that check the equality condition can be replaced by the switch case construct. It is also called fall through construct.

69. GENERAL FORMAT OF SWITCH STATEMENT

```
switch (variable or expression)
{
  case value1:   statement;
                 break;
  case value2 :  statement;
                 break;
  case valuen :  statement;
                 break;
  default:       statement;
                 break;
}
```

70. BREAK IN SWITCH CONSTRUCT

The break statement terminates the current case and transfers control outside the switch block.

71. TERNARY OPERATOR (? :)

An if-condition can also be written by using the ternary operator(?:).
Example :

```
   if (a != b)
   c = a;
else
   c = b;
```

Can be written as :

```
c = (a != b) ? a : b;
```

72. SYSTEM.EXIT()

This method in System class of java.lang package, exits current program by terminating ruming Java virtual machine. This method takes a status code. A non-zero value of status code is generally used to indicate abnormal termination.
exit(0) : Indicates successful termination.
exit(1) or exit(-1) : Indicates unsuccessful termination.

73. ITERATION

The process of repeating a set of instructions for a specific number of times or conditionally/ unconditionally.

74. LOOP

A loop executes a set of statements repetitively in a program.

Download INDIA's Best Study app NODIA

75. CONTROL VARIABLE

A variable initialized with a value and incremented or decremented as required in a loop is called a control variable. It controls the repetition of the loop body.
Example :

```
for (int i = 1; i < 10; i+2)
// i is the control variable.
```

76. COUNTER VARIABLE

A variable that stores an initial value and gets incremented by 1 when a condition is satisfied. It counts the number of times a condition is true.

General Format of for Construct:

(a) Single Statement : for (initializing expression; condition; control variable updation expression)
statement;

(b) Multiple Statements : for (initializing expression; condition; control variable updation expression)

```
{
 statement 1;
 statement 2;
}
```

(c) Infinite Loop :

```
 for (; condition;) // initializing and increment/decrement expressions ale omitted.
      statement; // // statement is repeated infinite times causing infinite loop
```

General Format of while Construct:

(i)

```
(initialization; condition or boolean expression)
    statement;
    updation;
```

(ii)

```
Statements :
initialization;
while (condition or boolean expression)
{
    statement 1;
    statement 2;
    updation;
}
```

General Format of do-while Construct:

```
initialization;
```

NODIA

```
do
{
    statement 1;
    statement 2;
    updation;
}  while  (condition  or  boolean
expression);
```

77. FINITE LOOP

A loop that executes a fixed number of times or until a condition becomes false.
Example:
for (int i = 0; i > 5; i++) { }

78. INFINITE LOOP

A loop that repeats indefinitely as there is no terminating condition.

79. DELAY

A for loop does not execute the loop body if a semi-colon (;) is added at the end of the loop. It updates the loop variable but skips the body.
Example:

```
for (i = 1; i <= 1000; i++);
System.out.print("Hello");
```

80. BREAK AS JUMPING STATEMENT

Used to transfer the flow of control to another part of a program. In Java, the break statement branches the execution and can terminate loops or switch cases. It can be used with labels to specify which loop to break.

81. BREAK TO EXIT A CONSTRUCT

In a switch construct, the break statement transfers control outside the loop or switch body.

82. CONTINUE STATEMENT

Changes the flow of execution by skipping the current iteration and moving to the next iteration of the loop.
General Format of continue : continue;

83. NESTED FOR LOOP CONSTRUCT

General Format :

```
for    (initializing    expression;
condition;       increment/decrement
expression) {
for    (initializing    expression;
condition;       increment/decrement
expression) {
        statement 1;
        statement 2;
    }
}
```

84. LABEL

A label is an identifier for a Java statement, used with a colon.

General Format of break with Label : break label;

85. LABELLED CONTINUE STATEMENT

A labelled continue skips the current iteration of an outer loop marked with the given label.

General Format of Labelled continue : continue label.

MULTIPLE CHOICE QUESTION

1. How many bytes of memory does the int data type occupy in Java ?
(a) 4 bytes
(b) 8 bytes
(c) 2 bytes
(d) 1 byte

Ans : MAIN 2023

(a) 4 bytes
The int data type in Java requires 4 bytes of memory and can store values ranging from -2^{31} to $2^{31}-1$.

2. What is the total number of primitive data types available in Java?
(a) 4
(b) 6
(c) 8
(d) 2

Ans : SQP 2022

(c) 8
Java provides 8 primitive data types, which are the basic building blocks for handling data.

3. Which operator has higher precedence in an expression containing both && and || in Java?
(a) &&
(b) ||
(c) Both have equal precedence
(d) None

Ans : COMP 2023

Downloaded from www.pdf.tube

(a) &&
In Java, the && (logical AND) operator has higher precedence than the || (logical OR) operator. This means that && will be evaluated first in an expression unless parentheses are used to explicitly define the order of operations.

4. In the statement t = t + (i + x);, which variable acts as the accumulator?
(a) t
(b) i
(c) x
(d) None

Ans : SQP 2021

(a) t
In this statement, t is the accumulator because it stores the cumulative result of the addition operation (t + (i + x)) after each iteration or operation. Accumulators are used to keep a running total or cumulative result.

5. Which of the following statements is valid for assigning a character in Java?
(a) Char c = '\t';
(b) Char c = '\abc';
(c) Char c = 'ghf';
(d) None of the above

Ans : MAIN 2012

(a) Char c = '\t';
Option (a) is valid because '\t' represents a valid escape sequence for the tab character in Java. The char data type can hold a single character or a valid escape sequence.

6. In Java, which package does the String class belong to?
(a) Java.io
(b) Java.string
(c) Java.lang
(d) Java.awt

Ans : SQP 2018

(c) java.lang
The String class is a part of the java.lang package, which is fundamental to the Java programming language. This package is automatically imported into every Java program, making the String class and other essential classes like Math and Object readily available.

Download INDIA's Best Study app NODIA

7. Which of the following statements is invalid in Java?
(a) int x, y;
(b) int x = 0, y = 0;
(c) int x, y = 0;
(d) long x, int y = 0;

Ans : MAIN 2020

(d) long x, int y = 0;
Option (d) is invalid because Java does not allow declaring variables of different types (e.g., long and int) in the same declaration statement. Each variable type must be declared separately.

NODIA

8. State the output of the following code:
int x = 3, y = 0; x *= 3 + y;
(a) 9
(b) 11
(c) 0
(d) 6

Ans : COMP 2019

(a) 9
The given code snippet uses the compound assignment operator *= This operator combines multiplication and assignment in a single step. The statement x*= 3 + y is equivalent to x = x* (3 + y). Initially, the variable x is assigned a value of 3, and y is assigned a value of 0. When the expression inside the parentheses, 3 + y, is evaluated, it results in 3 because y is 0. The compound assignment operator then multiplies the current value of x (which is 3) by the result of the expression (which is also 3). This results in x = 3 * 3, making the final value of x equal to 9. Therefore, the output of the code is 9.

9. Which of the following statements will result in an error in Java?
(a) Double x = 5f;
(b) Double x = 5.0;
(c) Float x = p3;
(d) Float x = 5.0;

Ans : SQP 2016

(c) Float x = p3
Option (c) will give an error because p3 is not defined or declared in the code. In Java, any variable used must be declared before it is assigned or referenced. Using an undeclared variable will result in a "cannot find symbol" error.

Download INDIA's Best Study app NODIA

10. What is the output of the following Java statement?
System.out.print("Hello" + "" + 2 + 1);
(a) Hello 3
(b) Hello 21
(c) Hello 2+1
(d) None of the above

Ans : MAIN 2011

(b) Hello 21
In Java, the + operator concatenates strings when one of the operands is a string. The statement starts with "Hello", which is concatenated with " " (empty string), resulting in "Hello". Next, 2 and 1 are treated as strings during concatenation, producing "Hello2" and then "Hello21". Thus, the output is "Hello21", and the correct answer is (b).

11. A mechanism where one class acquires the properties of another class:
(a) Polymorphism
(b) Inheritance
(c) Encapsulation
(d) Abstraction

Ans : MAIN 2010

(b) Inheritance
Inheritance allows a class to acquire the properties and methods of another class, promoting code reuse. It establishes a parent-child relationship. Other options (polymorphism, encapsulation, and abstraction) do not involve acquiring properties.

12. Which of the following is a characteristic of Java?
(a) Java is a low-level programming language.
(b) Java uses pointers for memory management.
(c) Java supports multi threading.
(d) Java does not support object-oriented programming principles.

Ans : SQP 2005

(c) Java supports multi threading.
Java supports multi threading, allowing multiple tasks to be performed simultaneously. It is a high-level, platform-independent, and object-oriented programming language that avoids explicit pointer usage for security.

13. What does the term "byte code" in Java refer to?
(a) Machine code executed directly by the CPU.
(b) Intermediate code produced by the Java compiler.
(c) Source code written in Java.
(d) Assembly code specific to hardware.

Ans : COMP 2007

(b) Intermediate code produced by the Java compiler.
Byte code is machine-independent intermediate code generated by the Java compiler, which the Java Virtual Machine (JVM) interprets into machine language.

14. Which of the following is NOT a principle of Object-Oriented Programming (OOP)?
(a) Encapsulation
(b) Polymorphism
(c) Iteration
(d) Inheritance

Ans : SQP 2009

(c) Iteration
Encapsulation, polymorphism, and inheritance are core principles of OOP, while iteration refers to looping constructs and is not related to OOP principles.

15. What is the primary purpose of the new operator in Java?
(a) To create a new variable.
(b) To allocate memory for an object at runtime.
(c) To terminate a program.
(d) To compare two objects.

Ans : COMP 2011

(b) To allocate memory for an object at runtime.
The new operator allocates memory for objects at runtime and returns a reference to the created object.

16. Which statement about Java's data types is true?
(a) The boolean data type can store more than two values.
(b) The char data type uses 1 byte.
(c) The float data type is used for single-precision numbers.
(d) The int data type stores characters.

Ans : MAIN 2015

(c) The float data type is used for single-precision numbers.

In Java, float is a 4-byte data type used for storing single-precision floating-point numbers. char uses 2 bytes, and boolean can only store true or false.

Click Here to Buy book on Amazon

17. What is the role of the break statement in a switch construct?

(a) It continues to the next iteration of a loop.

(b) It terminates the switch case and exits the construct.

(c) It skips the current iteration in a loop.

(d) It declares a variable inside a switch case.

Ans : MAIN 2019

(b) It terminates the switch case and exits the construct.

The break statement is used in switch constructs to terminate the current case and transfer control outside the switch block.

18. What does the term "encapsulation" mean in Java?

(a) Hiding the implementation details of a class and exposing only necessary functionality.

(b) Combining multiple classes into one.

(c) Allowing multiple methods to have the same name.

(d) Creating child classes from a parent class.

Ans : SQP 2003

(a) Hiding the implementation details of a class and exposing only necessary functionality.

Explanation : Encapsulation refers to the bundling of data and methods within a class, allowing controlled access through methods.

19. Which Java construct is used to repeat a block of statements until a condition becomes false?

(a) Selection construct

(b) Iteration construct

(c) Branching statement

(d) Jumping statement

Ans : COMP 2020

(b) Iteration construct

Iteration constructs like for, while, and do-while are used to repeat blocks of statements until a condition evaluates to false.

20. What is the output of the following statement?

```
int a = 10;

int b = 5;
int result = (a > b) ? a : b;

System.out.println(result);
```

(a) 10

(b) 5

(c) True

(d) False

Ans : MAIN 2005

(a) 10

The ternary operator checks the condition (a > b) which is true. Hence, the value of a (10) is assigned to result and printed.

NODIA

21. What does the Math.PI constant represent?

(a) The value of the square root of 2.

(b) The value of pi (ϖ).

(c) The base of natural logarithms (e).

(d) The value of infinity.

Ans : SQP 2017

(b) The value of pi (ϖ).

The Math.PI constant in Java represents the double value of ϖ (approximately 3.14159), which is a mathematical constant.

22. Which of the following is the correct way to read an integer from the console using the Scanner class?

A. Int num = scan.nextInt();

B. Int num = scan.readInt();

C. Int num = scan.inputInt();

D. Int num = nextInt(scan);

(a) Only A

(b) Only B

(c) Both A and B

(d) All the three

Ans : COMP 2015

(a)Only A

The correct way to read an integer using the Scanner class is scan.nextInt();. Options B, C, and D are syntactically incorrect.

23. Which of the following is the correct syntax to declare a class in Java?

A. Public class MyClass {}

B. Class MyClass () {}

C. MyClass : public class {}

D. Public class = MyClass {}

(a) Only A

(b) Only B

(c) Both A and C

(d) All the three

Ans : MAIN 2008

(a) Only A
The correct syntax to declare a class in Java is public class MyClass {}. Options B, C, and D have incorrect syntax.

24. Which of the following is the correct way to create a multi-line comment in Java?
A. /* This is a multi-line comment */
B. // This is a multi-line comment //
C. <!-- This is a multi-line comment -->
D. ** This is a multi-line comment **
(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans : SQP 2016

(a) Only A
Explanation:
In Java, multi-line comments are enclosed between /* and */. Options B, C, and D are not valid comment formats in Java.

25. Which of the following is the correct way to use the Math class to find the square root of a number?
A. Double result = Math.sqrt(16);
B. Double result = Math.square(16);
C. Double result = sqrt.Math(16);
D. Double result = square.Math(16);
(a) Only A
(b) Only B
(c) Both A and C
(d) All the three

Ans : MAIN 2021

(a) Only A

The correct method to find the square root in Java using the Math class is Math.sqrt(16);. Options B, C, and D are invalid methods.

26. Which of the following is the correct way to represent a character in Java?
A. Char c = 'A';
B. Char c = "A";
C. Char c = A;
D. Char c = (char)"A";
(a) Only A
(b) Only B
(c) Both A and C
(d) All the three

Ans : COMP 2019

(a) Only A
In Java, a character is represented using single quotes, e.g., char c = 'A';. Options B, C, and D are invalid as they do not follow the correct syntax for character representation.

27. Rearrange the following statements to describe the principle of inheritance in Object-Oriented Programming (OOP):
1. Inheritance enables code re usability.
2. A derived class inherits properties and methods from a base class.
3. Inheritance allows a class to acquire the attributes and methods of another class.
4. It is a key principle of Object-Oriented Programming.
(a) (3) (4) (1) (2)
(b) (4) (3) (2) (1)
(c) (4) (1) (3) (2)
(d) (3) (2) (1) (4)

Ans : SQP 2005

(b) (4) (3) (2) (1)
Inheritance is a key OP principle that allows code re usability by enabling one class to inherit from another. The correct sequence highlights its importance and functionality.

28. Rearrange the following statements to define the role of a class in Java:
1. A class serves as a blueprint for objects.
2. It defines the structure and behavior shared by all its objects.
3. A class is a user-defined data type.
4. Objects of a class represent real-world entities.
(a) (1) (4) (3) (2)
(b) (3) (1) (2) (4)
(c) (1) (2) (4) (3)
(d) (4) (2) (3) (1)

Ans : MAIN 2008

(b) (3) (1) (2) (4)
The correct sequence explains the concept of a class starting from its definition as a user-defined data type to its use as a blueprint for creating objects.

Download INDIA's Best Study app NODIA

Downloaded from www.pdf.tube

29. Rearrange the following statements to describe polymorphism in Object-Oriented Programming:

1. Polymorphism allows the same function to behave differently based on input or context.
2. It is one of the four pillars of Object-Oriented Programming.
3. Polymorphism is implemented using method overloading and overriding.
4. The term means "many forms."

(a) (4) (2) (1) (3)
(b) (2) (4) (1) (3)
(c) (1) (3) (4) (2)
(d) (3) (2) (4) (1)

Ans : COMP 2003

(b) (2) (4) (1) (3)
The sequence starts by defining polymorphism as an OP principle, followed by its meaning and its implementations through overloading and overriding.

30. Rearrange the following statements to explain the concept of the break statement in Java:

1. It is often used in loops or switch statements.
2. The break statement is used to exit a loop or switch block prematurely.
3. When encountered, the break statement terminates the current block.
4. It is commonly referred to as a "jumping statement."

(a) (2) (1) (3) (4)
(b) (4) (3) (2) (1)
(c) (2) (3) (1) (4)
(d) (3) (4) (2) (1)

Ans : SQP 2009

(c) (2) (1) (3) (4)

The correct sequence introduces the break statement, explains its purpose, how it works, and its categorization as a jumping statement.

31. Rearrange the following statements to define the term "byte code" in Java:

1. Byte code is produced by the Java compiler.
2. It is machine-independent and can run on any platform with a JVM.
3. Byte code is an intermediate code between source code and machine code.
4. The Java Virtual Machine (JVM) interprets byte code into machine-executable code.

(a) (1) (3) (4) (2)
(b) (3) (1) (2) (4)
(c) (1) (2) (3) (4)
(d) (2) (1) (4) (3)

Ans : MAIN 2012

(b) (3) (1) (2) (4)

The sequence explains the concept of byte code, starting with its definition, its generation, platform independence, and interpretation by the JVM.

ASSERTION AND REASON

32. **Assertion (A) :** Java is a platform-independent programming language.
Reason (R) : Java programs are compiled into byte code, which is executed by the Java Virtual Machine (JVM).

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2022

(a) Both A and R are true, and R is the correct explanation of A.
Java's platform independence comes from its byte code, which can run on any JVM, making R the correct explanation of A.

33. **Assertion (A) :** Object-Oriented Programming (OP) is focused on procedures and logic.
Reason (R) : OP uses concepts like encapsulation, inheritance, and to model real-world entities.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2021

(d) A is false, but R is true.
OOP focuses on modeling real-world entities using encapsulation, inheritance, and polymorphism, making A incorrect and R true.

NODIA

34. **Assertion (A) :** The switch construct in Java is also known as a fall-through construct.
Reason (R) : The break statement is mandatory after every case in a switch construct.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : SQP 2023

(c) A is true, but R is false.
The switch construct is called a fall-through construct because without break, control moves to the next case. However, break is not mandatory, making R false.

35. **Assertion (A) :** Unicode characters in Java require 2 bytes of storage.
Reason (R) : Unicode is a standard that assigns unique numeric values to characters valid across platforms.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : COMP 2018

(a) Both A and R are true, and R is the correct explanation of A.
Unicode standard ensures cross-platform validity of characters, and Java uses 2 bytes to store them, making both A and R true with R correctly explaining A.

36. **Assertion (A) :** A nested for loop in Java allows one loop to exist inside another.
Reason (R) : The outer loop must always complete all iterations before the inner loop can execute.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : MAIN 2020

(c) A is true, but R is false.
A nested for loop allows one loop inside another, but the inner loop executes for each iteration of the outer loop. Hence, R is false.

37. **Assertion (A) :** Encapsulation is one of the key principles of Object-Oriented Programming (OP).
Reason (R) : Encapsulation allows data to be hidden within an object, ensuring controlled access through methods.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : COMP 2017

(a) Both A and R are true, and R is the correct explanation of A.
Encapsulation ensures that object data is accessible only through specific methods, which is a core principle of OP.

38. **Assertion (A) :** The keyword new in Java is used to allocate memory for an object at runtime.
Reason (R) : The new keyword returns the reference (address) of the allocated memory.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : SQP 2021

(a) Both A and R are true, and R is the correct explanation of A.
The new keyword dynamically allocates memory and returns its reference, making R the correct explanation for A.

39. **Assertion (A) :** The Scanner class is used to take user input in Java.
Reason (R) : The Scanner class supports multiple data types, such as int, double, and char.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : COMP 2023

Click Here to Buy book on Amazon

(a) Both A and R are true, and R is the correct explanation of A.
The Scanner class allows for reading input in various data types, making R the correct explanation of A.

40. **Assertion (A) :** The byte data type in Java is capable of storing large numerical values.
Reason (R) : The range of the byte data type is from -128 to +127.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : SQP 2020

(d) A is false, but R is true.
The byte data type has a small range of -128 to +127, making A false and R true.

41. **Assertion (A) :** Java uses ASCII to store characters.
Reason (R) : ASCII assigns unique binary values to characters and symbols, allowing computers to store and process them.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2017

(d) A is false, but R is true.
Java uses Unicode, not ASCII, to store characters, making A false. However, R correctly explains ASCII.

TWO MARK QUESTIONS

42. What is object-oriented programming?

Ans : SQP 2023

Object-oriented programming simulates the real world in software. The real world consists of objects.

43. Name two Process Oriented languages and two Object Oriented languages.

Ans : MAIN 2022

Two Process Oriented Languages are : C and Fortran
Two Object Oriented Languages are : C++ and Java

44. Name any two principles/characteristics of Object Oriented Programs.

Ans : MAIN 2023

Some basic principles of object-oriented programming are Encapsulation, Abstraction, Inheritance, and Polymorphism.

45. Write the usage of different OPs concepts.

Ans : COMP 2021

Encapsulation : Leads to abstraction and protects the object from misuse.
Abstraction : Prevents the object from mishandling and misuse.
Inheritance : Derived classes inherit features of the base class, avoiding redefinition.
Polymorphism : Allows method/function overloading with the same name but different behaviors.

46. Define the term Byte code.

Ans : COMP 2020

Java programs are both compiled (translating all instructions of the source program to machine language at one go) and interpreted (translating instructions one by one). The Java compiler translates source code to an intermediate level language called byte code, which is independent of the machine and can run on any system.

47. What is the need of JVM?

Ans : SQP 2019

To run a Java program written on another system, its bytecode needs to be transferred to its own Java Virtual Machine. JVM is an interpreter which translates the program to specific machine language and runs it to give the result.

48. Explain the compilation process/execution stages in Java.

Ans : MAIN 2015

Compilation Process/Execution Stages in Java:
Java programs are both compiled and interpreted. The Java compiler translates the source code to an intermediate level language called bytecode. It is independent of the machine on which the

Downloaded from www.pdf.tube

program is run. The same byte code file can be used by any system.
To run a Java program written on another system, its byte code needs to be transferred to its own Java Virtual Machine. JVM is an interpreter which translates the program to its own machine language and runs it to give the result.

49. What features of Java have made it so popular that a need has been felt to introduce it at beginners' level?

Ans : COMP 2011

Features of Java making it so popular:
(a) It is a purely object-oriented programming language.
(b) It can be used to build distributed applications i.e., data or resources can be accessed from any networked computer easily.
(c) Java programs are both compiled and interpreted.
(d) Java programs are portable, i.e., can be carried to different machines in the form of byte code.
(e) Java programs can handle unexpected faults and errors.
(f) Java performs type checking on the user machine. On the net, while the programs are transferred, any changes occurring in the program are trapped as errors.
(g) It allows performing many tasks at the same time.

50. Why is Java called Internet language?

Ans : COMP 2020

Java programs are translated to Java byte code. Since bytecode is not affected by the machine environment, it is easier to run these programs on a wide range of machines. Only a Java interpreter (JVM) is required to be implemented on the machines. JVM, along with the Java Application Interface libraries, acts as a platform for the Java programs to run on the machine. All JVMs interpret the same byte code. This makes the program easier to be transported to a variety of machines, thus making it portable.
The worldwide web on the Internet demands portable programs, such as Java, to be distributed on the Internet. The second reason for its popularity as an Internet language is that it concentrates on real-world objects.

Download INDIA's Best Study app NODIA

51. Differentiate between a class and an object.

Ans : MAIN 2022, 2016

Class	Object
Class is a blueprint or template from which objects are created.	Object is an instance of a class.
Class has logical existence.	Object has physical existence.
For a class, memory space is not allocated when it is created.	In the case of an object, memory space is allocated when it is created.
Class is declared once.	Object is created many times.
Class is declared using the keyword 'class'.	Object is created through the keyword 'new'.

NODIA

52. Why is a class called an object factory?

Ans : SQP 2014

Each class describes possibly an infinite set of individual objects. So, it can be called a factory of objects. For example, a class of cellphones can have numerous objects such as Nokia, Samsung, and Panasonic, etc. The features and functions of the objects are described within the class.

53. How to create an object in Java?

Ans : COMP 2009

An object is created from a class. In Java, the new keyword is used to create a new object.
For example :
prog1 obj1 = new prog1();
Where prog1 is the class, obj1 is the newly created object's name, and prog1() is the constructor of the class.

54. Why is an object called "an instance of a class"?

Ans : MAIN 2005

A class encapsulates the data members and member functions. It is used to create an object. A class is an object factory. Hence, an object is called an instance of a class.

55. Why is a class known as composite data type?

Ans : SQP 2013

A class is called a composite data type because it binds up (encapsulates) one or more primitive types together as a single data type.
Example:

```
class Abc {
    int x;
    float y;
}
```

56. What is the difference between Unicode and ASCII?

Ans : MAIN 2021

Unicode is a superset of ASCII, and the numbers 0–128 have the same meaning in ASCII as they have in Unicode. For example, the number 65 represents capital "A."

ASCII	Unicode
It uses 7 or 8 bits of 1-byte space reserved for it. Sometimes one last bit is used for another significance.	It uses 2 Bytes (16 bits) in Java, encoding 65536 characters. The latest version can cover almost 1 Lakh characters.
It is meant for English letters only.	It can be utilized for most of the languages in the world.

57. What are the escape sequences? List the known sequences with their functions.

Ans : SQP 2004

Some characters, if preceded by a backslash (\), change their meaning and perform a specific function in Java. These are known as escape sequences.

Escape Sequences	Non-graphic Characters
\\	Backslash
\"	Double quotes
\n	New line
\t	Horizontal tab
\r	Carriage return
\f	Form feed

58. Write the memory capacity (storage size) of short and float data type in bytes.

Ans : MAIN 2018

Size of short data type = 2 Bytes
Size of float data type = 4 Bytes

59. What does the token 'keyword' refer to, in the context of Java? Give an example of keyword.

Ans : COMP 2009

A keyword is a reserved word in Java that has a special meaning. A keyword cannot be used as an identifier in the program. Examples of keywords : main(), int, if, for.

60. Name the keyword which:
(i) Indicates that a method has no return type.
(ii) Makes the variable as a class variable.

Ans : SQP 2005

(i) void keyword indicates that the method has no return type.
(ii) static keyword makes the variable a class variable.

61. What are variables?

Ans : COMP 2002

A variable is a memory location that stores a value, identified by a variable name. This stored value may change during the execution of the program, which is why it is called "variable.

62. What do you mean by 'data type' of a variable?

Ans : MAIN 2011

Data type refers to the type of value to be stored in a variable or the type of value to be used for calculations and other manipulations in expressions.

63. Write the range of acceptable values of these data types:
(i) Int
(ii) Short
(iii) Long
(iv) Byte

Ans : COMP 2022

(i) Int : -2^{31} to $2^{31}-1$
(ii) Short : -32,768 to 32,767
(iii) Long : -2^{63} to $2^{63}-1$
(iv) Byte : -128 to +127 (signed)

64. What is typecasting? Write an example of each kind of typecasting.

Ans : MAIN 2013

Click Here to Buy book on Amazon

Typecasting is a process of data type conversion when a variable of one data type is assigned to another.
Example of implicit typecasting:
When a character is stored in an integer variable:

```
int x = 'A';
System.out.print(x);
Output : 65
```

65. (i) Which type of typecasting are shown in the following examples?
(a) Double x = 15.2; int y = (int)x;
(b) Int x = 12; long y = x;
(ii) State the difference between a boolean literal and a character literal.

Ans : SQP 2013

(i) (a) Explicit typecasting
(b) Implicit typecasting
(ii)

Boolean Literal	Character Literal
Boolean literals are true and false.	A character literal is any character enclosed within single quotes.
Stored in variables with the boolean data type.	Stored in variables with the char data type.

66. What is unary operator? Discuss various unary operators used in Java.

Ans : MAIN 2019

Unary operators are operators that deal with only one operand (a single variable or a literal for an arithmetic unary operator and a single relational expression for a unary logical operator). Some of the unary operators used in Java are:

1. Logical NOT operator (!) : Performs logical negation on an expression.
2. Increment operator (++) : Used to increase the value of a variable by 1.
3. Decrement operator (--) : Used to decrease the value of a variable by 1.

Types of Increment and Decrement Operators:

1. Postfix operator : Changes after the action (e.g., p++, p--).
2. Prefix operator : Changes before use (e.g., ++p, --p).

67. What are the binary operators?

Ans : MAIN 2020

Binary operators take two operands to perform an operation.

1. Arithmetic operators : +, -, *, /, %
2. Relational operators : <, >, ==, <=, >=, !=
3. Logical operators : &&, ||

68. What is a ternary operator? Explain the ternary operator used in Java language.

Ans : COMP 2004

Operators that deal with three operands are called ternary operators. The conditional operator (?:) is a ternary operator used in Java.
String fare = (age < 60) ? "Full charge" : "Half charge";

69. Differentiate between Primitive Data Types and Non-Primitive Data Types:

Ans : SQP 2007, 2001

Primitive Data Types	Non-Primitive Data Types
Independent data types.	Dependent on primitive data types.
It has predefined types of data.	User has to define before using them.
It holds actual value.	It holds the address of the value.
It has fixed memory.	Its memory varies.
Example : byte, int, long, double, etc.	Example : array, classes, interfaces, etc.

70. Operators with higher precedence are evaluated before operators with relatively lower precedence. Arrange the operators given below in order of higher precedence to lower precedence.
(i) &&
(ii) %
(iii) >=
(iv) ++

Ans : SQP 2017

Order of operator precedence from higher to lower:
++, %, >=, &&

71. (i) Find the output of the following program segment : When:

NODIA

(a) val = 500
(b) val = 1600

```
int val, sum, n = 550;
sum = n + val > 1750 ? 400 : 200;
System.out.println(sum);
```

(ii) If the value of basic = 1500, what will be the value of tax after the following statement is executed?

```
Tax = basic > 1200 ? 200 : 100;
```

Ans : COMP 2008

(i) (a) 200
(b) 200
(ii) tax = 200

72. Write down Java expression for:
$T = \sqrt{(A^2 + B^2 + C^2)}$

Ans : MAIN 2014

```
Double T = Math.sqrt(A*A + B*B + C*C);
```

73. Identify the literals listed below:
(i) 0.5
(ii) 'A'
(iii) false
(iv) "a"

Ans : COMP 2023

(i) 0.5 – Floating literal
(ii) 'A' – Character literal
(iii) false – Boolean literal
(iv) "a" – String literal

74. If int y = 10, then find

```
int z = (++y * (y++ + 5));
```

Ans : MAIN 2010

z = 176

75. What are the values of x and y when the following statements are executed?

```
int a = 63, b = 36;
boolean x = (a > b) ? true : false;
int y = (a < b) ? a : b;
```

Ans : COMP 2003

x = true
y = 36

76. Give one point of difference between unary and binary operators.

Ans : MAIN 2017

Unary operators operate on one operand (e.g., ++, --).

Binary operators operate on two operands (e.g., +, -).

77. Write a Java expression for $\sqrt{2as + u^2}$

Ans : SQP 2019

```
double z = Math.sqrt(2 * a * s + u * u);
```

78. State one difference between the floating point literals : float and double.

Ans : MAIN 2002

Float	Double
A float is a single-precision data type whose size is 4 bytes.	A double is a double-precision data type whose size is 8 bytes.

79. Write the Java expressions for: $\frac{a^2 + b^2}{2ab}$

Ans : SQP 2000

```
double x = (a * a + b * b) / (2 * a * b);
```

80. (i) What will be the result stored in x after evaluating the following expression?

```
int x = 4;
x += (x++) + (4 + 4 + x) + x;
```

(ii) If the value of x = 2, y = 3, and z = 1, find:

```
v = x + --z + y++ + y;
```

Ans : SQP 2009

(i) x = 4 + 4 + 4 + 6 + 6 = 20
(ii) v = 2 + 0 + 3 + 4 = 9

81. Name the type of error (syntax, runtime, or logical error) in each case given below:
(i) Division of a variable by a variable that contains a value of zero.
(ii) Use of multiplication operator for a division operation.
(iii) Missing a semicolon in a Java sentence.

Ans : MAIN 2016

(i) Runtime error
(ii) Logical error
(iii) Syntax error

82. What is meant by precedence of operators?

Ans : COMP 2012

The order in which the operators are evaluated in an expression is called the precedence of operators or the hierarchy of operators. For

Downloaded from www.pdf.tube

example, multiplication and division have a higher precedence than addition and subtraction.

83. Rewrite the following decision-making statements using the conditional operator:

(i)

```
if (x > 0)
   flag = 1;
else
   flag = 0;
```

(ii)

```
if (x <= 0)
  value = 2 * x;
if (x > 0 && x < 10)
   value = 5 * x;
else
   value = 0;
```

Ans : COMP 2017

(i)

```
flag = (x > 0) ? 1 : 0;
```

(ii)

```
value = (!(x <= 0)) ? ((x > 0 && x < 10) ? (5 * x) : 0) : (2 * x);
```

84. (i) What is the use and syntax of a ternary operator?

(ii) Rewrite the given expression : (.r>0) ? (a=0) : (D=0) using if-else statement;

(iii) Rewrite the following program segment using if-else statements instead of the ternary operator.

```
String grade = (mark >= 90) ?
"A" : (mark >= 80) ? "B" : "C";
```

(iv) State the values of n and ch,

```
char c = 'A';
int n = c + 1;
char ch = (char) n;
```

Ans : MAIN 2023

(i) Syntax of ternary operator : Condition? expression to be evaluated when condition is true (:) expression tobe evaluated when condition is false.
Use : It can be used in place of an if-else-statement to make decisions and then to-evaluate the expressions selectively.

(ii)

```
if (x > 0)
   a = 0;
else
   b = 0;
```

(iii)

```
if (mark >= 90)
   grade = "A";
  else if (mark >= 80)
   grade = "B";
  else
 grade = "C";
```

(iv)

n = 66 , ch = 'B'

85. Find the value of n, if the value of p = 5, q = 19:

```
int n = (q - p) > (p - q) ? (q - p)
: (p - q);
```

Ans : SQP 2001

n = 14

86. Write a Java expression for ut + ½ ft².

Ans : MAIN 2009

```
double s = u * t + 1.0 / 2 * f * t
* t;
```

87. What will be the output of the following code?

```
int k = 5, j = 9;
k += k++ - ++j + k;
System.out.println("k=" + k);
System.out.println("j=" + j);
```

Ans : COMP 2018

k = 6, j = 10

88. Write an expression in Java for :
$z = (5x^3 + 2y) / (x + y)$

Ans : SQP 2007

```
double z = (5 * Math.pow(x, 3) + 2 * y) / (x +
y);
```

89. (i) What is the result stored in x after evaluating the following expression?

```
int x = 5;
x = x++ * 2 + 3 * --x;
```

(ii) What will be the output of the following code?

(a)

```
int m = 2;
int n = 15;
for (int i = 1; i < 5; i++)
m++;
--n;
```

NODIA

```
System.out.println("m=" + m);
System.out.println("n=" + n);
```

(b)

```
char x = 'A';
int m;
m = (x == 'a') ? 'A' : 'a';
System.out.println("m=" + m);
```

Ans : MAIN 2013

(i) x = 25
(ii) (a) m = 3 n = 14
(b) m = 97

90. Evaluate the following expression:

(i) If a = 5, b = 9, calculate the value of a = a + +-++b + a;.

(ii) If the value of x = 2, y = 3, and z = 1:
`v = x + --z + y++ + y;`

Ans : SQP 2020

(i) a = 6
(ii) v = 9

91. (i) What will be the output of the following, if x = 5 initially?
(a) 5 * ++x
(b) 5 * x++

(ii) What is the output of the following?

```
char c = 'A';
short m = 26;
int n = c + m;
System.out.println(n);
```

Ans : COMP 2021, 2014

(i) (a) 30
(b) 25
(ii) n = 91

92. State the data type and value of res after the following is executed:

```
char ch = '9';
res = Character.isDigit(ch);
```

Ans : MAIN 2017

char ch = '9';
res = Character.isDigit(ch);

93. Is a++ equivalent to ++a ?

Ans : SQP 2020

The two statements, if written independently, increase the value of a by 1.
Example

```
int a = 10;
a++;
System.out.println(a); // Result : 11
int a = 10;
++a;
System.out.println(a); // Result : 11
```

Both statements will result in 11 as the value of a.

94. What is the difference in pre-increment, post-increment, pre-decrement, and post-decrement?

Ans : MAIN 2005

(i) Pre-increment operator (++):
Applied before the operand. It increments the value of the operand by 1 and then performs other operations.
Example:

```
int x = 5;
System.out.println(++x);
// Output : 6
```

(ii) Post-increment operator (++) : Applied after the operand. It first performs associated operations and then increments the operand value by 1.

```
int x = 2;
System.out.println(x++);
// Output : 2
```

(iii) Pre-decrement (--) and Post-decrement (--):
These work similarly to pre-increment and post-increment, respectively, with the difference being that they decrease the operand value by 1.

95. What are the types of casting shown by the following examples?

(i)

```
double x = 15.2;
int y = (int) x;
```

(ii)

```
int x = 12;
long y = x;
```

Ans : SQP 2013

(i) Explicit typecasting
(ii) Implicit typecasting

96. Name the package that contains the Scanner class.

Ans : COMP 2004

java.util

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

97. Name the class that is used to read formatted input.

Ans : MAIN 2008

Scanner

98. Name the methods of the Scanner class that:

(i) Is used to input an integer data from the standard input device.

(ii) Is used to input a String data from the standard input device.

Ans : SQP 2000

(i) nextInt()
(ii) next()

99. What is the difference between next() and next(). charAt(0) methods of the Scanner class?

Ans : MAIN 2019

(i) str.next() returns a String input, which needs to be assigned to a string variable.
(ii) You cannot assign a String directly to a char.
(iii) However, you can assign the first character of the String to a char using charAt(0).

Example:

```
char ch = str.next().charAt(0);
```

100. How can you write comments in a Java program?

Ans : SQP 2011

(i) Single-line comments can be added using //. Everything after // is treated as a comment.
(ii) Multi-line comments start with /* and end with */.

Example:

```
int num = 5; /* This initializes num
as 5 */
```

101. Which of the following are valid comments?

(i) /* comment */
(ii) /* comment
(iii) // comment
(iv) */ comment */

Ans : MAIN 2006

(i) and (iii) are valid comments.

102. What is meant by a package? Name any two Java Application Programming Interface packages.

Ans : SQP 2022

A package is a names pace (like a folder) that contains a set of related classes together.
Examples of Java API packages:

(i) java.io
(ii) java.lang

103. What are User-defined packages?

Ans : COMP 2018

When a programmer needs to group his own classes that he has designed for his applications, he can create his own package using the package keyword followed by a package name. These are called user-defined packages.

104. Name the API package that contains wrapper classes.

Ans : SQP 2015

Java.lang

105. Name the package that automatically gets included in the Java source code (program) without any explicit import statement.

Ans : MAIN 2009

Java.lang

106. Which package has defined the class System?

Ans : COMP 2001

The System class is defined in the java.lang package.

107. In which package are the two classes String and StringBuffer defined?

Ans : SQP 2021

Java.lang package

108. What is the output of the following program?

```
double b = -15.6;
double a = Math.rint(Math.abs(b));
System.out.println("a=" + a);
```

Ans : MAIN 2014

A = 16.0

109. What are library classes? Give an example.

Ans : SQP 2019

Library classes are pre-defined Java classes that are included in an application program automatically. These classes include built-in member variables, constructors, and methods like ordinary classes.
Example : System, Math.

110. What will the following functions return when executed?

NODIA

(i) Math.max(-17, -19)
(ii) Math.ceil(7.8)

Ans : SQP 2007

(i) -17
(ii) 8.0

111. Give the output of the following program segment:

```
double x = 2.9, y = 2.5;
System.out.println(Math.min(Math.
floor(x), y));
System.out.println(Math.max(Math.
ceil(x), y));
```

Ans : MAIN 2013

```
Math.min(Math.floor(x), y) = 2.5
Math.max(Math.ceil(x), y) = 3.0
```

112. What are Java API packages?

Ans : COMP 2009

Java API groups a large number of classes into various packages according to their functions. These packages can be imported into a program as needed.

113. What are the final values stored in variables x and y below?

```
double a = -6.35;
double b = 14.74;
double x = Math.abs(Math.ceil(a));
double y = Math.rint(Math.max(a, b));
```

Ans : MAIN 2004

x = 6.0
y = 15.0

114. Write down Java expressions for:

(i) $T = \sqrt{A^2 + B^2 + C^2}$

(ii) $|x^2 + 2xy|$

Ans : SQP 2002

(i)

```
T = Math.sqrt(A * A + B * B + C * C);
```

(ii)

```
double d = Math.abs(x * x + 2 * x *
y);
```

115. Give the output of the following Math functions:

(i) Math.ceil(4.2)
(ii) Math.ceil(4.2)

Ans : SQP 2019

(i) 5.0
(ii) 4

116. Write the output of the below ceil and floor methods:

Code	Output
Math.ceil(5.9)	
Math.ceil(5.2)	
Math.ceil(-5.9)	
Math.ceil(-5.2)	
Math.ceil(0)	
Math.ceil(9.0)	
Math.floor(5.9)	
Math.floor(5.2)	
Math.floor(-5.9)	
Math.floor(-5.2)	
Math.floor(0)	
Math.floor(9.0)	

NODIA

Ans : MAIN 2017

Code	Output
Math.ceil(5.9)	6.0
Math.ceil(5.2)	6.0
Math.ceil(-5.9)	-5.0
Math.ceil(-5.2)	-5.0
Math.ceil(0)	0.0
Math.ceil(9.0)	9.0
Math.floor(5.9)	5.0
Math.floor(5.2)	5.0
Math.floor(-5.9)	-6.0
Math.floor(-5.2)	-6.0
Math.floor(0)	0.0
Math.floor(9.0)	9.0

117. Write the output of the below methods:

(a) Double x = Math.round(8.9);
(b) Double x = Math.round(8.1);
(c) Double x = Math.round(-8.9);
(d) Double x = Math.round(-8.1);
(e) Double x = Math.round(-8.0);

Ans : COMP 2007

(a) 9.0
(b) 8.0

(c) -9.0
(d) -8.0
(e) -8.0

118. What is a block?

Ans : MAIN 2020

A block is a set of instructions enclosed in braces. It begins with an opening curly brace { and ends with a closing curly brace }.

119. Rewrite the below using a switch-case construct:

```
 if (ch == 'Y') {
    x = x - 10;
    System.out.println(x);
} else if (ch == 'y') {
    x += 10;
    System.out.println(x);
} else {
    System.out.println("STOP");
}
```

Ans : SQP 2009

```
switch(ch) {
    case 'Y':
        x = x - 10;
        System.out.println(x);
        break;
    case 'y':
        x += 10;
        System.out.println(x);
        break;
    default:
        System.out.println("STOP");
        break;
}
```

120. Rewrite the below using a conditional operator:

```
if (discount > 500) {
          System.out.println("Rs"  +
discount);
} else {
          System.out.println("Rs"  +
(discount + 500));
}
```

Ans : SQP 2004

```
System.out.println((discount > 500) ?
("Rs" + discount) : ("Rs" + (discount
+ 500)));
```

121. Distinguish the if-else construct from a switch-case construct.

Ans : MAIN 2016

Distinguishing characteristics of if-else construct from a switch-case construct are:

(i) if-else allows conditions to compare a variable with a range of values, whereas switch-case can only compare for equality with specific values.
(ii) if-else can compare floating-point values, but switch-case is limited to non-floating types like integers, characters, and booleans.
(iii) if-else allows logical expressions in conditions; switch-case does not.
(iv) if-else can compare variables, but switch-case compares only to constants.

122. What is the significance of a break statement in a switch-case construct?

Ans : SQP 2020

When a variable's value matches the constant in a case, the statements in that case are executed until a break is encountered.

(i) If there is no break, the execution continues through subsequent case blocks.
(ii) A break acts as a terminator for the current case and prevents fall-through to other case blocks.

123. State the output of the following program segments according to the inputs as mentioned with each part:

(i)

```
if (year % 100 == 0) {
    if (year % 400 == 0)
            System.out.println("LEAP
YEAR");
    else
        System.out.println("NOT LEAP
YEAR");
} else {
System.out.print('NOT   A   CENTURY
YEAR");;
}
(i) year = 2051
(ii) year = 2000
(iii) year = 2171
(iv) year = 1900
```

(ii)

```
if (++x == 11) {
    System.out.print("RESULT IS 11");
} else if (++x == 10) {
    System.out.print("RESULT IS 10");
} else if (--x == 10) {
```

Downloaded from www.pdf.tube

ICSC CLASS 9

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Language

English Literature

Physics

Chemistry

Biology

Mathematics

Geography

History and Civics

Economics

Commercial Studies

Environmental Science

Physical Education

Computer Application

Economics Application

Commercial Application

Environ. Application

Home Science

Hindi

Robotics and AI

Yoga

For Other Subject PDF of all Classes, and for NEET, JEE and CUET visit

www.pdf.tube

ICSC CLASS 10

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Language	English Literature	Physics	Chemistry
Biology	Mathematics	Geography	History and Civics
Economics	Commercial Studies	Environmental Science	Physical Education
Computer Application	Economics Application	Commercial Application	Environ. Application
Home Science	Hindi	Robotics and AI	Yoga

BUY IT ON amazon

DOWNLOAD FREE PDF WWW.MCQGPT.COM

For Other Subject PDF of all Classes, and for NEET, JEE and CUET visit

www.pdf.tube

ISC CLASS 11

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Com

Mathematics

Physics

Chemistry

Biology

Physical Education

Environmental Science

Computer Science

Accountancy

Business Studies

Economics

Commerce

Legal Studies

Mass Media

Home Science

History

Geography

Political Science

Psychology

Sociology

For Other Subject of All Clasess PDF and for NEET, JEE and CUET visit

www.pdf.tube

ISC CLASS 12

Previous Year Solved Exam Question Bank

Click your Subject and Instantly get a free PDF or Order the Hard copy from Amazon.

English Com

Mathematics

Physics

Chemistry

Biology

Physical Education

Environmental Science

Computer Science

Accountancy

Business Studies

Economics

Commerce

Legal Studies

Mass Media

Home Science

History

Geography

Political Science

Psychology

Sociology

For Other Subject of All Clasess PDF and for NEET, JEE and CUET visit

www.pdf.tube

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

```
    System.out.print("RESULT IS 10");
} else {
        System.out.print("RESULT  IS
NEITHER 10 NOR 11");
}
(i) x = 9
(ii) x = 10
(iii) x = 11
(iv) x = 12
```

(iii)

```
 switch (num) {
    case 10:
    case 100 :
       System.out.println("original"
+ num);
        ++num;
    case 1000:
    case 500 :
        ++num;
    default :
        --num;
    System.out.print("Final" + num);
}
(i) num = 10
(ii) num = 100
(iii) num = 0
(iv) num = 500
```

(iv)

```
switch (a) {
    case 'a' :
        v += 1;
        break;
    case 'e' :
        v += 1;
    case 'i' :
        v += 1;
        break;
    case 'o' :
        v += 1;
        System.out.print(v + "'");
    case 'u' :
        v += 1;
        System.out.print(v);
        break;
    default :
       System.out.println("Character
is not a lower case vowel");
}
System.out.print("PROGRAMMING     IS
FUN");
(i) a stores character 'A'
(ii) a = 'o'
(iii) a = 'e'
(iv) a = 'u'
```

Ans : SQP 2020

(i) (a) NOT A CENTURY YEAR
(b) LEAP YEAR
(c) NOT A CENTURY YEAR
(d) NOT LEAP YEAR

(ii) (a) RESULT IS 10
(b) RESULT IS NEITHER 10 NOR 11
(c) RESULT IS 11
(d) RESULT IS NEITHER 10 NOR 11

(iii) (a) Original 10
Final 11
(b) Original 100
Final 101
(c) Final -1
(d) Final 500

(iv) (a) Character is not a lower case vowel
PROGRAMMING IS FUN
(b) 1 2 PROGRAMMING IS FUN
(c) PROGRAMMING IS FUN
(d) 1 PROGRAMMING IS FUN

124. Analyse the following program segment and determine how many times the loop will be executed and what will be the output of the program segment:

```
int p = 200;
while (true) {
    if (p < 100)
        break;
    p = p - 20;
}
System.out.println(p);
```

Ans : MAIN 2016

The loop will execute 6 times, and the output will be 80.

125. Analyse the following program segment and determine how many times the loop will be executed and what will be the output of the program segment:

```
int k = 1, i = 2;
while (++i < 6)
    k *= i;
System.out.println(k);
```

Ans : COMP 2010

The loop will execute 3 times, and the output will be 60.

126. (i) State one similarity and one difference between while and for loop.

(ii) Analyze the following program segment and determine how many times the body of the loop will be executed (show the working):

```
x = 5; y = 50;
while (x <= y) {
 y = y / x;
 System.out.println(y);
 }
```

(iii) Convert the following segment into an equivalent do-while loop:

```
int x, c;
for (x = 10, c = 20; c >= 10; c
= c - 2)
 x++;
```

Ans : SQP 2022

(i) **Similarity :** Both while and for are looping statements. The statements in the loop body are executed whenever the condition is satisfied.
Difference : A for statement is used when the number of iterations is known. A while statement is used when the number of iterations is not known but depends on some other condition (e.g., user input).

(ii) Working:

Iteration 1 : y = 50 / 5 = 10
Iteration 2 : y = 10 / 5 = 2
Iteration 3 : Condition 5 <= 2 is false.
The loop body is executed two times.

(iii)

```
int x, c;
x = 10; c = 20;
do {
  x++;
  c = c - 2;
} while (c >= 10);
```

127. What is meant by an infinite loop? Give an example.

Ans : MAIN 2012

An infinite loop executes the loop body repeatedly without termination. It never ends because there is no condition to break the loop, or the condition always evaluates to true.

Example:

```
for (i = 10; i > 5; i++) {
    System.out.print("Hello");
}
```

128. Write the output of the following program code:

```
char ch;
int x = 97;
do {
    ch = (char) x;
    System.out.print(ch + " ");
    if (x % 10 == 0)
        break;
    ++x;
} while (x <= 100);
```

Ans : SQP 2008

Output : a b c d

129. Convert the following segment into an equivalent for loop:

```
int i; i = 0;
while (i <= 20) {
    System.out.print(i + " ");
    i++;
}
```

Ans : MAIN 2003

```
for (int i = 0; i <= 20; i++) {
    System.out.print(i + " ");
}
```

130. How many times will the following loop execute? What value will be returned?

```
int x = 2, y = 50;
do {
    ++x;
    y -= x + x++;
} while (x <= 10);
return y;
```

Ans : COMP 2013

(i) The loop will execute 5 times.
(ii) The value returned will be 15.

131. Rewrite the following program segment using while instead of for statement.

```
int f = 1, i;
for (i = 1; i <= 5; i++) {
    f *= i;
    System.out.println(f);
}
```

Ans : MAIN 2017

```
int f = 1, i;
```

Click Here to Buy book on Amazon

```
i = 1;
while (i <= 5) {
    f *= i;
    System.out.println(f);
    i++;
}
```

132. What is the purpose of using continue statement?

Ans : SQP 2010

A continue statement is used to stop the sequential flow of statements and skip to the next iteration of the loop.

133. Analyze the given program segment and answer the following questions:

Ans : COMP 2010

```
for (int m = 5; m <= 20; m += 5) {
    if (m % 3 == 0)
        break;
    else if (m % 5 == 0)
        System.out.println(m);
    continue;
}
```

134. (i) How many times does the body of the loop get executed?

(ii) Write the output of the program segment.

Ans : MAIN 2020

(i) The loop executes 3 times.

(ii) 5,10

135. Compare all the three types of loops?

Ans : MAIN 2016

	For loop	While loop	Do-while loop
Usage	It is used when the number of times the loop will execute is known in advance.	It is more suitable if the loop termination is not known beforehand.	It is a good choice when you are asking a question whose answer will determine if the loop is repeated or terminated.
Control Type	It is an entry control loop, as the condition is checked before executing the loop.	It is preferred when you don't want to execute the loop body before checking, even once.	When you want to execute the loop body at least once, it is an exit control loop.

Download INDIA's Best Study app NODIA

136. Write the general format of for loop.

Ans : SQP 2001

```
for(initializing expression;
condition; update expression) {
    statements;
}
```

NODIA

137. What is the difference between a break statement and a continue statement when they occur in a loop?

Ans : COMP 2004

(i) A break statement terminates the sequential execution of the statements in the loop or switch construct and transfers the control to the statements outside the loop/switch.

(ii) A continue statement skips the sequential execution of the remaining statements in the loop and continues with the next iteration, after updating the loop variable.

Example 1 : Using break

```
for(int i = 2; i <= 10; i = i + 2) {
    if(i == 8) break;
    System.out.print(i + " ");
}
Output : 2 4 6
```

Example 2

```
for(int i = 2; i <= 10; i = i + 2) {
    if(i == 8) continue;
    System.out.print(i + " ");
}
Output : 2 4 6 10
```

138. What is the difference between break and labelled break?

Ans : MAIN 2016

A break statement has two forms : labeled and unlabeled.

The unlabeled break terminates the innermost loop or switch construct where it resides.

A labeled break allows the termination of outer loops even if the break statement resides in an inner loop.

139. **(i)** Write the output of the below statements:

```
for (int f = 10; f >= 5; f--) {
  int x = -5;
  for (int g = 1; g <= 2; g++) {
      x = x + 1;
    }
  System.out.println(x);
    }
```

(ii)

```
int x = 1;
outer:
for (int i = 1; i <= 5; i++) {
   inner:
  for (int j = 10; j <= 50; j++) {
     if (j % 2 != 0)
         x = x + 1;
         break outer;
     else
         break inner;
   }
  }
  System.out.print(x);
```

Ans : COMP 2021

(i) The program prints -3 six times in different lines:
-3
-3
-3
-3
-3
-3

(ii) The output is 1.

FIFTEEN MARKS QUESTIONS

140. Using a switch statement, write a menu-driven program to convert a given temperature from Fahrenheit to Celsius and vice versa. For an incorrect choice, an appropriate error message should be displayed.

$$C = \frac{5}{9} \times (F - 32) \text{ and } F = 1.8 \times C + 32$$

Ans : SQP 2023

```
import java.util.*;
class Temp {
    public static void main() {
        String t;
     Scanner buf = new Scanner(System.
in);
        System.out.println("Menu:");
              System.out.println("1.
Fahrenheit to Celsius");
      System.out.println("2. Celsius
to Fahrenheit");
           System.out.println("Enter
your choice:");
        int c = buf.nextInt();

        switch (c) {
            case 1:
           System.out.println("Enter
the temp in Fahrenheit:");
              int f = buf.nextInt();
                 double cl = 5.0 / 9
* (f - 32);
             System.out.println("The
temp in Celsius = " + cl);
                break;

            case 2:
           System.out.println("Enter
the temp in Celsius:");
                   double c2 = buf.
nextDouble();
                double f2 = 1.8 * c2
+ 32;
             System.out.println("The
temp in Fahrenheit = " + f2);
                break;

            default:
             System.out.println("The
choice is incorrect.");
        }
    }
}
```

141. Using the switch statement, write a menu-driven program for the following:
To print Floyd's triangle (Given below):
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

Ans : MAIN 2022

```
import java.util.*;
// Program for Floyd's Triangle
public class pyr {

    public static void floyd() {
        int l, c, n = 1;
        for (l = 1; l <= 5; l++) {
            System.out.println();
           for (c = 1; c <= l; c++) {
                System.out.print(n +
" ");
                n++;
```

Downloaded from www.pdf.tube

```
            }
        }
    }

    public static void main(String[]
args) {
         System.out.println("Enter 1
for Floyd's triangle");
      Scanner sc = new Scanner(System.
in);
        int i = sc.nextInt();
        switch (i) {
            case 1:
                floyd();
                break;

            case 2:
                            System.out.
println("Sorry, wrong choice");
        }
    }
}
```

142. Accept a number from the keyboard and check whether it is a positive number, negative number, or zero. Print suitable messages.

Ans : SQP 2022

```
public class Positive {
    public static void main(int num)
{
        if (num > 0)
            System.out.print(num + "
is a positive number");
        else if (num == 0)
            System.out.print(num + "
is ZERO");
        else
            System.out.print(num + "
is a negative number");
    }
}
```

143. Write a program to check if a number is odd or even. Print suitable messages for the number.

Ans : COMP 2023

```
public class Ev {
    public static void main(int num)
{
        if (num % 2 == 0)
            System.out.print(num + "
is an even number.");
        else
            System.out.print(num + "
is an odd number.");
    }
}
```

144. Write a program to check if a number is odd or even. Print suitable messages for the number.

Ans : MAIN 2021

```
public class Ev {
    public static void main(int num)
{
        if (num % 2 == 0)
            System.out.print(num + "
is an even number.");
        else
            System.out.print(num + "
is an odd number.");
    }
}
```

145. An electricity board charges the bill depending on the number of units consumed as follows:

(i) First 100 units : 40 P per unit

(ii) Next 200 units : 60 P per unit

(iii) Above 300 units : 1 per unit

Write a Java program to print the net bill to be paid by a consumer. The net bill includes a rent of 250.00 charged from a consumer.

Ans : COMP 2021

```
public class Electricity {
     public static void main(String
nam, long pv, long ps) {
          // Taking customer's name,
previous reading, and present reading
inputs as function parameters
        long units = ps - pv;
        double charge = 0;

        if (units <= 100)
            charge = units * 0.40;
        else if (units <= 300)
              charge = 100 * 0.40 +
(units - 100) * 0.60;
        else
           charge = 100 * 0.40 + 200
* 0.60 + (units - 300) * 1;

         double netCharge = charge +
250;
        System.out.println();
         System.out.println("\tDELHI
VIDYUT BOARD ELECTRICITY BILL");
```

NODIA

```
System.out.println
}ln("*****************************
******");
        System.out.println("Name : "
+ nam);
        System.out.println("Previous
units : " + pv);
      System.out.println("Net Charge
: " + netCharge);
    }
}
```

146. Write a program that returns the commission in Rupees for a salesman.
The commission is based on the following conditions:

Sales	Commission %
< 5000	5%
5000 to 10000	8%
> 10000	10%

Ans : SQP 2014

```
public class Commission {
     public static void main(double
sales) {
        double c = 0;

        if (sales < 5000)
            c = 5.0 / 100 * sales;
          else if (sales >= 5000 &&
sales <= 10000)
            c = 8.0 / 100 * sales;
        else
            c = 10.0 / 100 * sales;

        System.out.println(c);
    }
}
```

147. Write a class Leap to accept a year and check if it is a leap year or not.

Ans : COMP 2020

```
import java.util.*;
public class Leap {
    public static void checkleap() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter a year to check : ");
        int yr = buf.nextInt();
        if (yr % 4 == 0)
          System.out.print("The year
is a leap year.");
        else
          System.out.print("The year
is not a leap year.");
    }

     public static void main(String
args[]) {
        checkleap();
    }
}
```

148. Write a program to compute and display the roots of a quadratic equation of the form $ax^2 + bx + c$

(i) If b^2 - 4ac results in a negative answer, print IMAGINARY ROOTS

(ii) If b^2-4ac gives a positive result, print REAL AND DISTINCT ROOTS.

(iii) If b^2-4ac=0, print EQUAL ROOTS.

Ans : MAIN 2010

```
import java.util.*;

public class Quadratic {
     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the values of a, b, and c for
the quadratic equation:");
        int a = buf.nextInt();
        int b = buf.nextInt();
        int c = buf.nextInt();
        double root1, root2;

       if (b * b - 4 * a * c == 0) {
          System.out.println("EQUAL
ROOTS");
           root1 = -b / (2 * a);
         System.out.print("The root
value is = " + root1);
        } else if (b * b - 4 * a * c
< 0) {
         System.out.print("IMAGINARY
ROOTS");
        } else {
          root1 = (-b + Math.sqrt(b
* b - 4 * a * c)) / (2 * a);
          root2 = (-b - Math.sqrt(b
```

Click Here to Buy book on Amazon

```
* b - 4 * a * c)) / (2 * a);
            System.out.println("REAL
AND DISTINCT ROOTS");
             System.out.println("The
two roots are : " + root1 + " and "
+ root2);
        }
    }
}
```

149. A set of two linear equations with two unknown values x_1 and are x_2 given as:

(i) $m = ax_1 + bx_2$

(ii) $n = cx_1 + dx_2$

The solution of the equation is given as :

(i) $x_1 = \frac{md - bn}{ad - cb}$

(ii) $x_2 = \frac{na - mc}{ad - cb}$

Provided that the denominator $ad - cb$ is not equal to 0.

Write an appropriate message. Write a Java program that will read the values of a, b, c, d, m, and n and computes and displays the results x_1 and x_2.

Ans : SQP 2008

```
import java.util.*;

public class Linear {
     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);

          System.out.println("Please
enter the values of a, b, c, d, m and
n respectively");
        double a = buf.nextDouble();
        double b = buf.nextDouble();
        double c = buf.nextDouble();
        double d = buf.nextDouble();
        double m = buf.nextDouble();
        double n = buf.nextDouble();

        if ((a * d - c * b) != 0) {
            double x1 = (m * d - b *
n) / (a * d - c * b);
            double x2 = (n * a - m *
c) / (a * d - c * b);

             System.out.println("The
results are : " + x1 + " and " + x2);
        } else {
     System.out.println("Denominator
cannot be zero");
        }
    }
}
```

150. An employee is entitled to pay an income tax based on his gross annual income as follows:

Annual Gross Income	Annual Tax Deduction %
Less than or equal to ₹100000	0%
₹100001 to ₹500000	₹1000 + 10% of income exceeding ₹100000
₹500001 to ₹800000	₹5000 + 20% of income exceeding ₹500000
> ₹800000	₹10000 + 30% of income exceeding ₹800000

Write a Java program to compute and print the tax payable by a salaried person.

Ans : SQP 2004

```
import java.util.*;

class IncomeTax {
     public static void main(String
args[]) {
      Scanner A = new Scanner(System.
in);
       System.out.println("Enter the
gross income");
        int i = A.nextInt();
        double t;

        if (i <= 100000) {
            t = (0.0 / 100) * i;
             System.out.println("TAX
= " + t);
        } else if (i > 100000 && i <=
500000) {
             t = 1000 + (10.0 / 100)
* (i - 100000);
             System.out.println("TAX
= " + t);
        } else if (i > 500000 && i <=
800000) {
             t = 5000 + (20.0 / 100)
* (i - 500000);
             System.out.println("TAX
= " + t);
        } else if (i > 800000) {
            t = 10000 + (30.0 / 100)
```

Downloaded from www.pdf.tube

```
* (i - 800000);
                System.out.println("TAX
= " + t);
          }
     }
}
```

151. Define a Java class Maximum to print the largest out of the three numbers accepted from the user.

Ans : MAIN 2013

```
import java.util.*;

class Maximum {
     public static void main(String
args[]) {
      Scanner A = new Scanner(System.
in);
       System.out.println("Enter the
first no.:");
        double a = A.nextDouble();
       System.out.println("Enter the
second no.:");
        double b = A.nextDouble();
       System.out.println("Enter the
third no.:");
        double c = A.nextDouble();

        if (a > b && a > c)
             System.out.println("The
largest no. is = " + a);
        else if (b > a && b > c)
             System.out.println("The
largest no. is = " + b);
        else
             System.out.println("The
largest no. is = " + c);
    }
}
```

152. WAP in Java to input a character from the keyboard and detect whether it is an alphabet or a digit. If it is an alphabet, check whether it is a lowercase alphabet or an uppercase alphabet. Print appropriate messages.

Ans : COMP 2005

```
import java.util.*;

public class Q12 {
     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Enter a
character:");
       char x = buf.next().charAt(0);
            boolean v = Character.
isLetter(x);

        if (v) {
       System.out.println("Character
entered is an alphabet");
        if (Character.isLowerCase(x))
                          System.out.
println("Character entered is a
lowercase alphabet");
            else
                          System.out.
println("Character entered is an
uppercase alphabet");
     } else if (Character.isDigit(x))
{
       System.out.println("Character
entered is a digit");
        } else {
       System.out.println("Character
entered is neither an alphabet nor a
digit");
        }
    }
}
```

153. Write a Java program that displays the result of the following evaluations based on the user's entry of choice from the keyboard:

(i) Natural logarithm of a number.

(ii) Absolute value of a number.

(iii) Square root of a number.

(iv) Random number between 0 and 1.

Ans : MAIN 2009

```
import java.util.*;
import java.lang.Math;

public class Q13 {
     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
       System.out.println("**** MENU
****");
       System.out.println("1. Natural
Logarithm");
      System.out.println("2. Absolute
Value");
       System.out.println("3. Square
```

NODIA

```
Root");
        System.out.println("4. Random
Number between 0 and 1");
         System.out.println();
           System.out.println("Please
enter your choice:");
         int ch = buf.nextInt();

         switch (ch) {
             case 1:
                          System.out.
println("Please enter a number whose
logarithm is to be found:");
                    double n = buf.
nextDouble();
              double x = Math.log(n);
             System.out.println("The
logarithm of the number is : " + x);
                break;
             case 2:
                          System.out.
println("Please enter a number to
find its absolute value:");
                    double a = buf.
nextDouble();
             System.out.println("The
absolute value is : " + Math.abs(a));
                break;
             case 3:
                          System.out.
println("Please enter a number to
find its square root:");
                    double s = buf.
nextDouble();

              System.out.println("The
square root is : " + Math.sqrt(s));
                break;
             case 4:
                double random = Math.
random();
              System.out.println("The
random number between 0 and 1 is : "
+ random);
                break;
             default:
                          System.out.
println("Invalid choice!");
         }
    }
}
```

154. Write a program that accepts an alphabet and checks whether it is a vowel or a consonant.

Ans : MAIN 2018

```
public class Q14 {
    public static void main(char x) {
         if (x == 'A' || x == 'a' ||
x == 'E' || x == 'e' || x == 'I' ||
x == 'i' || x == 'O' || x == 'o' ||
x == 'U' || x == 'u')
                System.out.print("The
entered character is a vowel");
        else
                System.out.print("The
entered character is a consonant");
    }
}
```

155. Design a class Library, define a method for the class that accepts the memberId, memberName, bookName, and computes the fine for a member. The fine is calculated depending on the number of days a book is returned late:

Days	Fine per day in Rupees
First Ten days	₹1
Next Ten days	₹5
Later than 20 days	₹8

Ans : SQP 2021

```
import java.util.*;

public class Library {
     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the member id, name and the
book name issued by him:");

        int x = buf.nextInt(); // id
          buf.nextLine(); // consume
the newline character
          String n = buf.nextLine();
// name
          String b = buf.nextLine();
// book

          System.out.println("Please
enter the number of days by which the
book is returned late:");
        int days = buf.nextInt();
        int fine;
```

Click Here to Buy book on Amazon

```
        if (days <= 10) {
            fine = days * 1;
        } else if (days <= 20) {
          fine = 10 + 5 * (days - 10);
        } else {
            fine = 10 + 5 * 10 + 8 *
(days - 20);
        }

         System.out.print("The fine =
" + fine);
    }
}
```

156. Write a program in Java to compute the railway fare depending on the criteria as given below:

Age (in years)	Distance (in Kms)	Fare (in Rupees)
Below 10	Below 10	₹ 5
	Between 10 and 50	₹ 20
	Above 50	₹ 50
Between 10 and 60	Below 10	₹ 10
	Between 10 and 50	₹ 40
	Above 50	₹ 80
Above 60	Below 10	₹ 4
	Between 10 and 50	₹ 15
	Above 50	₹ 35

Ans : COMP 2003

```
import java.util.*;

public class Railway {
     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);

          System.out.println("Please
enter your age in years:");

        int age = buf.nextInt();

          System.out.println("Please
enter the distance to be travelled:");
        int dis = buf.nextInt();

        if (age <= 10) {

            if (dis <= 10)
              System.out.print("Fare
= ₹5.00");
            else if (dis <= 50)
              System.out.print("Fare
= ₹20.00");
            else
              System.out.print("Fare
= ₹50.00");
         } else if (age > 10 && age
<= 60) {
            if (dis <= 10)
              System.out.print("Fare
= ₹10.00");
            else if (dis <= 50)

              System.out.print("Fare
= ₹40.00");
            else
              System.out.print("Fare
= ₹80.00");
        } else {
            if (dis <= 10)
              System.out.print("Fare
= ₹4.00");
            else if (dis <= 50)
              System.out.print("Fare
= ₹15.00");
            else
              System.out.print("Fare
= ₹35.00");
        }
    }
}
```

157. WAP to compute the amount that a customer pays for the taxi that he hires based on the following conditions:

Kms Travelled	Amount per Km
Upto less or equal to 1 Km	₹25
More than 1 to 6 Km	₹10
More than 6 to 12 Km	₹15
More than 12 to 18 Km	₹20
More than 18 Km	₹25

Input the customer's name, the taxi number, and the number of kilometers travelled by him.

Ans : SQP 2009

```
public class TaxiMeter {
     public static void main(String
name, int Km, int taxino) {
        double amt = 0;
```

NODIA

```
        if (Km <= 1)
            amt = 25 * Km;
        else if (Km > 1 && Km <= 6)
           amt = 25 + 10 * (Km - 1);
        else if (Km > 6 && Km <= 12)
            amt = 25 + 10 * 5 + 15 *
(Km - 6);
       else if (Km > 12 && Km <= 18)
            amt = 25 + 10 * 5 + 15 *
6 + 20 * (Km - 12);
        else
            amt = 25 + 10 * 5 + 15 *
6 + 20 * 6 + 25 * (Km - 18);

       System.out.println("Taxi No.\
tName\tKilometers    travelled\tBill
Amount");
          System.out.println(taxino
+ "\t" + name + "\t" + Km + "\t" +
amt);
    }
}
```

158. WAP to compute the amount that a customer pays for the taxi that he hires based on the following conditions:

Kms Travelled	Amount per Km
First 10 kms	₹25
Next 20 kms	₹10
Next 40 kms	₹15
Above 70 kms	₹12

Input the customer's name, the taxi number, and the number of kilometers travelled by him.

Ans : MAIN 2018

```
public class TaxiMeter {
     public static void main(String
name, int km, int taxino) {
        double amt = 0;
        if (km <= 10) // First 10 km
            amt = 25 * km;
       else if (km > 10 && km <= 30)
// Next 20 km
            amt = 25 * 10 + 10 * (km
- 10);
       else if (km > 30 && km <= 70)
// Next 40 km
             amt = 25 * 10 + 10 * 20
+ 15 * (km - 30);
        else if (km > 70)
             amt = 25 * 10 + 10 * 20
+ 15 * 30 + 12 * (km - 70);

       System.out.println("Taxi No.\
tName\tKilometers    travelled\tBill
Amount");
          System.out.println(taxino
+ "\t" + name + "\t" + km + "\t" +
amt);
    }
}
```

159. Using the switch statement, write a menu-driven program to:

(i) Generate and display the first 10 terms of the Fibonacci series 0,1,1,2,3,5,....
The first two Fibonacci numbers are 0 and 1 and each subsequent number is the sum of the previous two.

(ii) Find the sum of the digits of an integer that is input.
Sample Input : 15390
Sample Output : Sum of the digits = 18

(iii) For an incorrect choice, an appropriate error message should be displayed.

Ans : SQP 2020

```
import java.util.Scanner;

public class Ans4 {
    public static void main(String[]
args) {
     Scanner ob = new Scanner(System.
in);
     System.out.println("1. Generate
first 10 terms of Fibonacci series");
       System.out.println("2. Sum of
digits of an integer");
          System.out.println("Enter
your choice:");
        int ch = ob.nextInt();

        switch (ch) {
            case 1 :
                int a = 0, b = 1, c;
                System.out.print(a +
", " + b); // Two terms of Fibonacci
printed
                for (int i = 1; i <=
10 - 2; i++) { // Loop for remaining
8 terms
                    c = a + b;
                System.out.print(",
```

Downloaded from www.pdf.tube

```
" + c);
                    a = b;
                    b = c;
                }
               System.out.println();
// Line break after series
                break;

            case 2:
                        System.out.
println("Enter a number whose digits
will be summed:");
               int n = ob.nextInt();
                int sum = 0;
                while (n > 0) {
                    sum += n % 10;
                    n /= 10;
                }
             System.out.println("Sum
of the digits = " + sum);
                break;

            default:
                        System.out.
println("Invalid choice! Please try
again.");
        }
    }
}
```

160. Write a menu-based program that outputs the results of the following evaluations based on the number entered by the user:

(i) Absolute value of the number

(ii) Square root of the number

(iii) Random numbers between 0 and 1

Ans : COMP 2004

```
import java.util.*;

public class User {
     public static void main(String
args[]) {
      Scanner sc = new Scanner(System.
in);

           System.out.println("*****
MENU *****");
      System.out.println("1. Absolute
value");
       System.out.println("2. Square
Root");
       System.out.println("3. Random
number between 0 and 1");
        System.out.println();
          System.out.println("Please
enter your choice:");
        int ch = sc.nextInt();

        switch (ch) {
            case 1:
                        System.out.
println("Please enter a number whose
Absolute value is to be found:");
              int n1 = sc.nextInt();
               int m = Math.abs(n1);
             System.out.println("The
absolute value of the result is : "
+ m);
                break;

            case 2:
                        System.out.
println("Please enter a number whose
square root is to be found:");
              int n2 = sc.nextInt();
                  double m2 = Math.
sqrt(n2);
             System.out.println("The
square root of the result is : " +
m2);
                break;

            case 3:
                        System.out.
println("Random number between 0 and
1:");
                  double t = Math.
random();
              System.out.println(t);
                break;

            default:
                        System.out.
println("Invalid choice");
        }
    }
}
```

161. Write a menu-driven program to allow the user to enter a choice:

(i) Find the volume of a cuboid

(ii) Find the volume of a cube

Ans : MAIN 2001

```
import java.util.Scanner;
```

NODIA

```
class Volume {
    public static void main(String[]
args) {
        System.out.print("Press 1 to
find volume of cuboid");
        System.out.print("Press 2 to
find volume of cube");
        System.out.print("Enter your
choice : ");

     Scanner sc = new Scanner(System.
in);
        int ch = sc.nextInt();

        switch (ch) {
            case 1:
             System.out.print("Enter
the side of the cuboid : ");
              int side = sc.nextInt();
                  double vol = side *
side * side;
            System.out.print("Volume
of this cuboid is " + vol);
                break;

            case 2:
             System.out.print("Enter
the side of the cube : ");
                side = sc.nextInt();
                vol = side * side;
            System.out.print("Volume
of this cube is " + vol);
                  System.exit(0); //
Terminate the program

            default:
                          System.out.
print("You have entered an invalid
choice number");
        }

      System.out.print("This program
also demonstrated the use of System.
exit()");
    }
}
```

162. Write a program to input a number and print whether the number is a special number or not. (A number is said to be a special number if the sum of the factorial of the digits of the number is the same as the original number.)Example:
145 is a special number because 1!+4!+5!=1+24+120=1451!+4!+5!=1+24+120=145.
(Where ! stands for the factorial of the number, and the factorial value of a number is the product of all integers from 1 to that number, e.g., 5!=1×2×3×4×5=120).

Ans : COMP 2008

```
public class Q22 {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
         System.out.println("Enter a
number to test:");
        int n = sc.nextInt();
        int num = n;
        int r = 0, p, s = 0;

       // Separate the digits of the
number
          // Repeatedly divide the
number by 10 and get the remainder
        do {
            r = n % 10;
            n = n / 10;

           // Factorial of the digit
stored in r
            p = 1;
             for (int i = 1; i <= r;
i++) {
                p = p * i;
            }

            // Sum of the factorials
            s = s + p;
        } while (n > 0);

        if (num == s) {
           System.out.println("It is
a special number");
        } else {
           System.out.println("It is
not a special number");
        }
    }
}
```

Variables used:
num, n: Number to check.
r: Remainder that is the digit in the number.

Click Here to Buy book on Amazon

p: Stores the factorial of the digit.
s: Stores the sum of the factorials of the digits.

163. Write a menu-driven program as:
(i) To accept a number and check and display whether it is a prime number or not.
(ii) To check whether a number is an automorphic number or not (using switch-case statement).

Hint:
(i) Prime number : A number is said to be a prime number if it is divisible only by 1 and itself and not by any other number.
Example : 3, 5, 7, 11, 13, etc.
(ii) Automorphic number : An automorphic number is the number that is contained in the last digit(s) of its square.
Example : 25 is an automorphic number as its square is 625, and 25 is present as the last two digits.

Ans : SQP 2014

```
import java.util.*;

public class Number {
    public static void auto(int n) {
          // Function to check for
automorphic number
        int m = n;
        // Count number of digits in
the number
        int c = 0, r;
        do {
            r = n % 10;
            n = n / 10;
            c++;
        } while (n > 0);

        // Find square of the number
        int s = m * m;

       /* Extract c number of digits
from the square s and form a number
          from the extracted digits.
This number would be the reverse. */
        int rev = 0;
       for (int i = 1; i <= c; i++) {
            r = s % 10;
            rev = rev * 10 + r;
            s = s / 10;
        }

      /* Reverse the reversed number
to get the number to compare
         with the original number */
        // Convert to String
             String nn = Integer.
toString(rev);
         // Convert to StringBuffer
to reverse
            StringBuffer st1 = new
StringBuffer(nn);
        st1.reverse();
         // Convert original number
also to String
     String c2 = Integer.toString(m);
        String c1 = st1.toString();

        if (c1.compareTo(c2) == 0)
            System.out.println("The
inputted number is automorphic");
        else
            System.out.println("The
inputted number is not automorphic");
    }

   public static void prime(int n) {
      // Function to check for prime
number
        // A prime number has only
two factors : 1 and itself
       // Find the factors and count
them
        int c = 0;
       for (int i = 1; i <= n; i++) {
            if (n % i == 0)
                c++;
        }
        if (c == 2)
            System.out.println("It's
a prime number");
        else
            System.out.println("It's
not a prime number");
    }

    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
        System.out.println("MENU");
        System.out.println("1. Check
prime number");
```

```
        System.out.println("2. Check
automorphic number");
       System.out.println("Enter the
choice 1 or 2:");
        int ch = sc.nextInt();
           System.out.println("Enter
a number to check for prime or
automorphism:");
        int num = sc.nextInt();

        switch (ch) {
            case 1:
                prime(num);
                break;
            case 2:
                auto(num);
                break;
            default:
                              System.out.
println("Invalid choice, please
enter choice as 1 or 2");
        }
    }
}
```

164. Write a menu-driven program to:

(i) Accept a number and check whether it is a 'BUZZ' number or not.

(ii) Accept any two numbers and print the 'GCD' of them.

Hint:

(i) BUZZ number : A number is said to be a BUZZ number if it either ends with 7 or is divisible by 7.

(ii) GCD (Greatest Common Divisor) : The GCD of two integers is calculated using the continued division method. Divide the larger number by the smaller; the remainder then divides the previous divisor. The process is repeated until the remainder is zero. The divisor then becomes the GCD.

Ans : SQP 2019

```
import java.util.*;

class Menu {
    public static void Buzz(int num)
{
        if (num % 7 == 0 || num % 10
== 7)
            System.out.println("It is
a Buzz number");
        else
            System.out.println("It is
not a Buzz number");
    }

    public static void GCD(int a, int
b) {
        int dividend = 0, divisor = 0;
        if (a > b) {
            dividend = a;
            divisor = b;
        } else {
            dividend = b;
            divisor = a;
        }
        int r = 0;
        do {
            r = dividend % divisor;
            if (r != 0) {
                dividend = divisor;
                divisor = r;
            } else
                break;
        } while (true);

       System.out.println("GCD of " +
a + " and " + b + " is : " + divisor);
    }

    public static void main(String[]
args) {
     Scanner sca = new Scanner(System.
in);
        System.out.println("Menu");
         System.out.println("1. Buzz
Number");
       System.out.println("2. GCD");
           System.out.println("Enter
your choice 1 or 2:");
        int c = sca.nextInt();
        if (c == 1) {
           System.out.println("Enter
number to check:");
            int n = sca.nextInt();
            Buzz(n);
        } else {
           System.out.println("Enter
2 numbers:");
            int m = sca.nextInt();
            int n = sca.nextInt();
            GCD(m, n);
        }
    }
}
```

NODIA

Download INDIA's Best Study app NODIA

165. Write a program to calculate and print the sum of each of the following series:
(Value of x to be input by the user.)
(i) Sum (S) $= 2-4+6-8+\ldots-20$
(ii) Sum (S) $= \frac{x}{2}+\frac{x}{5}+\frac{x}{8}+\ldots\ldots+\frac{x}{20}$

Ans : COMP 2006

```
class Series {
    public static void series1() {
        int i, a = 2, sum = 0;
        for (i = 1; i <= 10; i++) {
            if (i % 2 == 0) {
                sum = sum - a;
            } else {
                sum = sum + a;
            }
            a = a + 2;
        }
            System.out.print("Sum of
series1 = " + sum);
    }

   public static void series2(double
x) {
        double sum = 0;
        for (int i = 2; i <= 20; i =
i + 3) {
            sum = sum + x / i;
        }
            System.out.print("Sum of
series2 = " + sum);
    }
}
```

Variables used:
i : Loop variable
sum : Sum of the series
x : Input variable

166. Write a program to calculate and print the sum of odd numbers and the sum of even numbers for the first n natural numbers. The integer n is to be entered by the user.

Ans : COMP 2020

```
import java.util.*;

class Sum {
    public static void main(String[]
args) {
        int so = 0, se = 0;
      Scanner b = new Scanner(System.
in);
           System.out.println("Enter
number:");
        int n = b.nextInt();

         // Loop for odd numbers up
to n
        for (int i = 1; i <= n; i +=
2) {
            so = so + i; // Find sum
of odd numbers
        }

         // Loop for even numbers up
to n
        for (int j = 2; j <= n; j +=
2) {
            se = se + j; // Find sum
of even numbers
        }

        System.out.print("Sum of odd
numbers = " + so);
       System.out.print("Sum of even
numbers = " + se);
    }
}
```

Variables used:
n : Limit of natural numbers up to which the sum is required.
so, se : Store the sum of odd and even natural numbers up to
n, respectively.
i, j : Loop variables.

167. Write a program to print all factors of a number n accepted from the user.

Ans : MAIN 2007

```
import java.util.*;

class Factors {
    public static void main(String[]
args) {
      Scanner A = new Scanner(System.
in);
       System.out.println("Enter the
value of a number whose factors are
to be printed:");
        int n = A.nextInt();
       for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                          System.out.
println("Factor = " + i);
            }
        }
    }
}
```

Downloaded from www.pdf.tube

168. Using the switch statement, write a menu-driven program:

(i) To find and display all the factors of a number input by the user (including 1 and excluding the number itself).
Example:
Sample Input:n=15
Sample Output : 1,3,5

(ii) To find and display the factorial of a number input by the user (the factorial of a non-negative integer n, denoted byn!, is the product of all integers less than or equal to n).
Example:
Sample Input : n=5
Sample Output : 5!=1×2×3×4×5=120
Note : For an incorrect choice, an appropriate error message should be displayed.

Ans : SQP 2016

```
import java.util.Scanner;

public class Menu {
    public static void main(String[]
args) {
      System.out.println("1. Factors
of number other than itself");
              System.out.println("2.
Factorial of a number");
          System.out.println("Enter
your choice : ");
    Scanner scan = new Scanner(System.
in);
        int ch = scan.nextInt();

        switch (ch) {
            case 1:
           System.out.println("Enter
the number:");
                    int num = scan.
nextInt();
                        System.out.
print("Factors : ");
                for (int i = 1; i <
num; i++) {
                 if (num % i == 0) {
                        System.out.
print(i + " ");
                    }
                }
                break;
            case 2:
          System.out.println("Enter
the number:");
             int n = scan.nextInt();
                int factorial = 1;
                for (int i = 1; i <=
n; i++) {
                    factorial *= i;
                }
                        System.out.
println("Factorial of " + n + " = "
+ factorial);
                break;

            default:
                        System.out.
println("Invalid choice. Please
enter 1 or 2.");
        }
    }
}
```

169. A special two-digit number is such that when the sum of its digits is added to the product of its digits, the result is equal to the original two-digit number.
Example:
Consider the number 59.
(i) Sum of digits = 5+9=14
(ii) Product of its digits = 5×9=45
(iii) Sum of the sum of digits and product of digits = 14+45=59
Write a program to accept a two-digit number. Add the sum of its digits to the product of its digits. If the value is equal to the number input, output the message "Special 2-digit number", otherwise output the message "Not a special 2-digit number".

Ans : MAIN 2002

```
import java.util.Scanner;

public class Special2DigitNumber {
    public static void main(String[]
args) {
          System.out.println("Please
enter a two-digit number");
    Scanner scan = new Scanner(System.
in);
        int num = scan.nextInt();

        int number = num;
       int sum1 = 0, sum = 0, p = 1,
```

NODIA

```
c = 0, r;

        do {
            r = num % 10;
            num = num / 10;
              sum = sum + r; // Add
digits
           p = p * r;     // Multiply
digits
             c++;            // Count
number of digits
        } while (num > 0);

        if (c != 2) {
         System.out.println("Invalid
number. Enter a two-digit number");
            System.exit(1);
        } else {
            sum1 = sum + p;
            if (sum1 == number)
                          System.out.
println("The number is a special two-
digit number");
            else
             System.out.println("The
number is not a special two-digit
number");
        }
    }
}
```

Variables used:

num : Number to check.
sum, sum1 : Sum variables.
p : Product of digits.
c : Counter for the number of digits in the number.

170. Write two separate programs to generate the following patterns using iteration (loop) statements:

(i)

```
*
# *
* # *
# * # *
* # * # *
```

(ii)

```
54321
5432
543
54
5
```

Ans : COMP 2020

(i)

```
public class PatternA {
    public static void main(String[]
args) {

        for (int i = 1; i <= 5; i++) {
             for (int j = 1; j <= i;
j++) {
                 if (j % 2 != 0)
                          System.out.
print("*");
                 else
                          System.out.
print("#");
             }
             System.out.println();
         }
    }
}
```

(ii)

```
public class PatternB {
    public static void main(String[]
args) {

        for (int i = 1; i <= 5; i++) {
             for (int j = 5; j >= i;
j--) {
                 System.out.print(j);
             }
             System.out.println();
         }
    }
}
```

171. WAP to compute the sum of the following series:

1. $(1+2)+(1+2+3)+(1+2+3+4)$
   $+\ldots+(1+2+3+\ldots+n)$
2. $\frac{1}{1+2}+\frac{1}{1+2+3}+\ldots+\frac{1}{1+2+3+..n}$
3. 1,4,9,16,25,…n terms.
4. 1,8,27,64,125,…n terms.
5. 0,7,26,63,124,…n terms.
6. 3,8,15,24,…n terms.
7. $\frac{x}{10}+\frac{x+2}{15}+\frac{x+4}{20}+\ldots..+\frac{x+n!}{m}$

Ans : SQP 2010

Click Here to Buy book on Amazon

1.

```
import java.util.*;

public class Q31a {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
          System.out.println("Please
enter the limit number : ");
        int n = obj.nextInt();
        int s = 1, sum = 0;
       for (int i = 2; i <= n; i++) {
            s += i;
            sum += s;
        }
        System.out.print("The sum of
the series is : " + sum);
    }
}
```

2.

```
import java.util.*;

public class Q31b {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int s = 1;
        float sum = 0;
       System.out.println("Enter the
value of n:");
        int n = obj.nextInt();
       for (int i = 2; i <= n; i++) {
            s += i;
            sum += (float) 1 / s;
        }
         System.out.print("The sum =
" + sum);
    }
}
```

3.

```
import java.util.*;

public class Q31c {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int y, s = 0, c = 1;
       System.out.println("Enter the
value of n:");
        int n = obj.nextInt();
       for (int i = 1; i <= n; i++) {
            y = i * i;
            s += y;
            c++;
        }
        System.out.println(s);
    }
}
```

NODIA

4.

```
import java.util.*;

class Q31d {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int s = 0, c = 1, y;
       System.out.println("Enter the
value of n:");
        int n = obj.nextInt();
       for (int i = 1; i <= n; i++) {
            y = i * i * i;
            s += y;
            c++;
        }
        System.out.println(s);
    }
}
```

5.

```
import java.util.*;

class Q31e {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int s = 0, c = 1, y;
       System.out.println("Enter the
value of n:");
        int n = obj.nextInt();
       for (int i = 1; i <= n; i++) {
            y = (i * i) - 1;
            s += y;
            c++;
        }
        System.out.println(s);
    }
}
```

6.

```
import java.util.*;

class Q31f {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int s = 0, c = 1, y;
       System.out.println("Enter the
value of n:");
        int n = obj.nextInt();
       for (int i = 2; c <= n; i++) {
            y = (i * i) - 1;
            s += y;
            c++;
        }
        System.out.println(s);
    }
}
```

7.

```
import java.util.*;

class Q31g {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        double s = 0, p = 1, sum = 0;
       System.out.println("Enter the
value of n:");
        int n = obj.nextInt();
       System.out.println("Enter the
value of x:");
        int x = obj.nextInt();
        int c = 10;
        for (int i = 0; i <= n; i +=
2) {
            for (int j = 1; j <= x +
i; j++) {
                p *= j;
            }
            s += p / c;
            c += 5;
        }
           System.out.print("Sum of
series = " + s);
    }
}
```

172. Write a program to accept a number and check and display whether it is a Niven number or not. (Niven number is that number which is divisible by its sum of digits).
Example:
Consider the number 126.
Sum of its digits is 1 +
2 + 6 = 9 and 126 is divisible by 9.

Ans : MAIN 2020

```
import java.util.*;

public class Q32 {
    public static void main() {
      Scanner sc = new Scanner(System.
in);
         System.out.println("Enter a
number ");
        int num = sc.nextInt();
        int m = num;
        int s = 0, p;

        while (num > 0) {
            p = num % 10;
            s = s + p;
            num = num / 10;
        }

         System.out.println("The sum
is " + s);
        if (m % s == 0)
            System.out.println(m + "
is a Niven number");
        else
            System.out.println(m + "
is not a Niven number");
    }
}
```

173. Write a program to accept a number and check and display whether it is a Niven number or not. (Niven number is that number which is divisible by its sum of digits).
Example:
Consider the number 126.
Sum of its digits is 1 +
2 + 6 = 9 and 126 is divisible by 9.

Ans : COMP 2013

```
import java.util.*;

public class Q32 {
```

Downloaded from www.pdf.tube

```
    public static void main() {
      Scanner sc = new Scanner(System.
in);
         System.out.println("Enter a
number ");
        int num = sc.nextInt();
        int m = num;
        int s = 0, p;

        while (num > 0) {
            p = num % 10;
            s = s + p;
            num = num / 10;
        }

         System.out.println("The sum
is " + s);
        if (m % s == 0)
            System.out.println(m + "
is a Niven number");
        else
            System.out.println(m + "
is not a Niven number");
    }
}
```

174. A tech number has an even number of digits. If the number is split into two equal halves, the square of the sum of these halves is equal to the number itself. Write a program to generate and print all four-digit tech numbers.
Example : Consider the number 3025
Square of the sum of halves $= (30 + 25)^2 = 55^2$
$= 3025$
3025 is a tech number.

Ans : COMP 2019

```
import java.util.*;

class TechNumber {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

         // Loop through all 4-digit
numbers
        for (int i = 1000; i <= 9999;
i++) {
            int q = i / 100; // First
two digits
             int r = i % 100; // Last
two digits
              int sum = q + r; // Sum
of halves
              int sq = sum * sum; //
Square of the sum

            // Check if the square of
the sum equals the number
            if (sq == i) {
                System.out.println(i
+ " is a tech number");
            }
        }
    }
}
```

NODIA

175. Design a class name ShowRoom with the following description:

Instance variables / Data members:

(i) String name - To store the name of the customer.

(ii) Long mobno - To store the mobile number of the customer.

(iii) Double cost - To store the cost of items purchased.

(iv) Double dis - To store the discount amount.

(v) Double amount - To store the amount to be paid after discount.

Member methods:

(i) ShowRoom() - Default constructor to initialize data members.

(ii) Void input() - To input customer name, mobile number, cost.

(iii) Void calculate() - To calculate discount on the cost of purchased items, based on the following criteria:

Cost Range	Discount (%)
Less than or equal to ₹ 10,000	5%
More than ₹ 10,000 and less than or equal to ₹ 20,000	10%
More than ₹ 20,000 and less than or equal to ₹ 35,000	15%
More than ₹ 35,000	20%

(iv) Void display() - To display customer name, mobile number, amount to be paid after discount.

Write a main method to create an object of the class and call the above member methods.

Ans : MAIN 2021

```
import java.util.*;

class ShowRoom {
    String name;
    double cost, dis, amount;
    long mobno;

      // Constructor to initialize
variables
    ShowRoom() {
        name = "";
        cost = 0.0;
        mobno = 0;
        dis = 0.0;
        amount = 0.0;
    }

    // Method to input details
    void input() {
     Scanner sc = new Scanner(System.
in);
          System.out.println("Please
enter customer name:");
        name = sc.nextLine();
           System.out.println("Enter
mobile number and cost:");
        mobno = sc.nextLong();
        cost = sc.nextDouble();
    }

     // Method to calculate discount
and amount
    void calculate() {
        if (cost <= 10000)
            dis = 0.05 * cost;
       else if (cost > 10000 && cost
<= 20000)
            dis = 0.10 * cost;
       else if (cost > 20000 && cost
<= 35000)
            dis = 0.15 * cost;
        else
            dis = 0.20 * cost;

        amount = cost - dis;
    }

    // Method to display details
    void display() {
        System.out.println("Customer
Name : " + name);
         System.out.println("Mobile
Number : " + mobno);
        System.out.println("Amount to
pay : ₹" + amount);
    }

    // Main method
    public static void main(String[]
args) {
        ShowRoom sr = new ShowRoom();
         sr.input();
         sr.calculate();
         sr.display();
    }
}
```

176. Write a program to generate and print all four-digit special numbers. A special number is defined as a number where the product of its two halves (when split into two equal parts) equals the original number.
Example : Consider the number 2025
Product of the halves = $(20)\times(25)=500$
500 is not equal to 2025, so it is not a special number.
Consider 3025 :
Product of halves = $(30)\times(25)=3025$
3025 is a special number.

Ans : MAIN 2022

```
import java.util.*;

class SpecialNumber {
    public static void main(String[]
args) {
         // Loop through all 4-digit
numbers
       for (int i = 1000; i <= 9999;
i++) {
           int q = i / 100; // First
two digits
            int r = i % 100; // Last
two digits
             int product = q * r; //
Product of the halves

             // Check if the product
of the halves equals the number
            if (product == i) {
                System.out.println(i
+ " is a special number");
            }
         }
    }
}
```

Click Here to Buy book on Amazon

177. Write a program to accept a number and check whether it is a Smith Number or not. (A Smith Number is a composite number whose sum of digits is equal to the sum of digits of its prime factors, excluding 1.)
Example:
Consider the number 85.
Prime factorization of 85 is 5×17.

Sum of digits of 85 =8+5=13.

Sum of digits of 5+17 =5+(1+7)=13.

Ans : SQP 2021

```
import java.util.*;

public class SmithNumber {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

         System.out.println("Enter a
number : ");
        int num = sc.nextInt();

        if (isPrime(num)) {
            System.out.println(num +
" is not a Smith number (it's a prime
number).");
            return;
        }

    int sumOfDigits = sumDigits(num);
       int sumOfPrimeFactorsDigits =
sumPrimeFactorsDigits(num);

          System.out.println("Sum of
digits : " + sumOfDigits);
             System.out.println("Sum
of digits of prime factors : " +
sumOfPrimeFactorsDigits);

                if (sumOfDigits ==
sumOfPrimeFactorsDigits) {
            System.out.println(num +
" is a Smith number.");
        } else {
            System.out.println(num +
" is not a Smith number.");
        }
    }

    // Method to check if a number
is prime
    public static boolean isPrime(int
n) {
        if (n <= 1) return false;
         for (int i = 2; i <= Math.
sqrt(n); i++) {
             if (n % i == 0) return
false;
        }
        return true;
    }

    // Method to calculate the sum of
digits of a number
     public static int sumDigits(int
n) {
        int sum = 0;
        while (n > 0) {
            sum += n % 10;
            n /= 10;
        }
        return sum;
    }

    // Method to calculate the sum of
digits of prime factors
              public   static   int
sumPrimeFactorsDigits(int n) {
        int sum = 0;
       for (int i = 2; i <= n; i++) {
            while (n % i == 0) {
                sum += sumDigits(i);
                n /= i;
            }
        }
        return sum;
    }
}
```

178. Write a menu-driven program to allow the user to enter a choice:

(i) Find the area of a rectangle.

(ii) Find the area of a square.

Ans : COMP 2024

```
import java.util.Scanner;

class AreaCalculator {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
```

Downloaded from www.pdf.tube

```
        // Display the menu
         System.out.println("Press 1
to find the area of a rectangle");
         System.out.println("Press 2
to find the area of a square");
        System.out.print("Enter your
choice : ");

        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                 // Find the area of
a rectangle
             System.out.print("Enter
the length of the rectangle : ");
                      double length =
sc.nextDouble();
             System.out.print("Enter
the width of the rectangle : ");
                       double width =
sc.nextDouble();
                double areaRectangle
= length * width;
                          System.out.
println("Area of the rectangle is :
" + areaRectangle);
                break;

            case 2:
                 // Find the area of
a square
             System.out.print("Enter
the side of the square : ");
                       double side =
sc.nextDouble();
                double areaSquare =
side * side;
            System.out.println("Area
of the square is : " + areaSquare);
                break;

            default:
                   // Handle invalid
choice
             System.out.println("You
have entered an invalid choice.");
        }

         System.out.println("Program
completed.");
    }
}
```

179. Write a menu-driven program to allow the user to enter a choice:

(i) Find the perimeter of a rectangle.

(ii) Find the perimeter of a square.

Ans : MAIN 2000

```
import java.util.Scanner;

class PerimeterCalculator {
    public static void main(String[]
args) {
      Scanner sc = new Scanner(System.
in);

        // Display the menu
       System.out.println("Press 1 to
find the perimeter of a rectangle");
         System.out.println("Press 2
to find the perimeter of a square");
        System.out.print("Enter your
choice : ");

        int choice = sc.nextInt();

        switch (choice) {
            case 1:
               // Find the perimeter
of a rectangle
             System.out.print("Enter
the length of the rectangle : ");
                      double length =
sc.nextDouble();
             System.out.print("Enter
the width of the rectangle : ");
                       double width =
sc.nextDouble();
           double perimeterRectangle
= 2 * (length + width);
                          System.out.
println("Perimeter of the rectangle
is : " + perimeterRectangle);
                break;

            case 2:
               // Find the perimeter
of a square
             System.out.print("Enter
the side of the square : ");
                       double side =
sc.nextDouble();
              double perimeterSquare
= 4 * side;
                          System.out.
```

NODIA

```
println("Perimeter of the square is
: " + perimeterSquare);
                break;

            default:
                  // Handle invalid
choice
             System.out.println("You
have entered an invalid choice.");
        }

         System.out.println("Program
completed.");
    }
}
```

180. Write a menu-driven program to allow the user to enter a choice:

(i) Convert temperature from Celsius to Fahrenheit.

(ii) Convert temperature from Fahrenheit to Celsius.

Ans : SQP 2023

```
import java.util.Scanner;

class TemperatureConverter {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Display the menu
         System.out.println("Press 1
to convert Celsius to Fahrenheit");
         System.out.println("Press 2
to convert Fahrenheit to Celsius");
        System.out.print("Enter your
choice : ");

        int choice = sc.nextInt();

        switch (choice) {
            case 1:
               // Convert Celsius to
Fahrenheit
             System.out.print("Enter
temperature in Celsius : ");
                    double celsius =
sc.nextDouble();
                 double fahrenheit =
(celsius * 9/5) + 32;
                          System.out.
println("Temperature in Fahrenheit :
" + fahrenheit);
                 break;

            case 2:
               // Convert Fahrenheit
to Celsius
             System.out.print("Enter
temperature in Fahrenheit : ");
                         fahrenheit =
sc.nextDouble();
                 celsius = (fahrenheit
- 32) * 5/9;
                          System.out.
println("Temperature in Celsius : "
+ celsius);
                 break;

            default:
                  // Handle invalid
choice
             System.out.println("You
have entered an invalid choice.");
        }

         System.out.println("Program
completed.");
    }
}
```

181. Write a menu-driven program to allow the user to enter a choice:

(i) Calculate the factorial of a number.

(ii) Check whether a number is prime or not.

Ans : COMP 2023

```
import java.util.Scanner;

class NumberOperations {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Display the menu
           System.out.println("Press
1 to calculate the factorial of a
number");
         System.out.println("Press 2
to check if a number is prime");
        System.out.print("Enter your
```

Click Here to Buy book on Amazon

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

```
choice : ");

        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                    // Calculate the
factorial of a number
                System.out.print("Enter
a number : ");
                int num = sc.nextInt();
                   long factorial = 1;

                   for (int i = 1; i <=
num; i++) {
                       factorial *= i;
                   }

                System.out.println("The
factorial of " + num + " is : " +
factorial);
                   break;

               case 2:
                   // Check if a number
is prime
                System.out.print("Enter
a number : ");
                   num = sc.nextInt();
                boolean isPrime = true;

                   if (num <= 1) {
                       isPrime = false;
                   } else {
                        for (int i = 2;
i <= Math.sqrt(num); i++) {
                            if (num % i
== 0) {
                                isPrime
= false;
                               break;
                           }
                       }
                   }

                   if (isPrime) {
                            System.out.
println(num + " is a prime number.");
                   } else {
                System.out.println(num
+ " is not a prime number.");
                   }
                   break;

               default:
                      // Handle invalid
choice
                System.out.println("You
have entered an invalid choice.");
           }

            System.out.println("Program
completed.");
       }
}
```

182. Write a menu-driven program to allow the user to enter a choice:

(i) Find the sum of digits of a number.

(ii) Reverse a number.

Ans : MAIN 2023

```
import java.util.Scanner;

class NumberOperations {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Display the menu
      System.out.println("Press 1 to
find the sum of digits of a number");
        System.out.println("Press 2
to reverse a number");
        System.out.print("Enter your
choice : ");

        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                 // Find the sum of
digits of a number
             System.out.print("Enter
a number : ");
              int num = sc.nextInt();
                 int sum = 0;

                 while (num > 0) {
                     sum += num % 10;
                     num /= 10;
                 }
```

NODIA

```
            System.out.println("The
sum of the digits is : " + sum);
                break;

            case 2:
                // Reverse a number
             System.out.print("Enter
a number : ");
                num = sc.nextInt();
                int reverse = 0;

                while (num > 0) {
                   reverse = reverse
* 10 + num % 10;
                    num /= 10;
                }

             System.out.println("The
reversed number is : " + reverse);
                break;

            default:
                  // Handle invalid
choice
             System.out.println("You
have entered an invalid choice.");
        }

         System.out.println("Program
completed.");
    }
}
```

www.nodia.in

Downloaded from www.pdf.tube

CHAPTER 2

CLASS AS THE BASIS OF ALL COMPUTATIONS

SUMMARY

1. CLASS

A class is a blueprint representing a group of objects with similar properties. It encapsulates all information (attributes and methods) required for its objects into a single unit, a concept known as encapsulation. This encapsulation aids in information hiding, also referred to as abstraction.

2. CLASS FEATURES IN OOP

Classes implement features of OOP, including:
(i) Abstraction
(ii) Inheritance
(iii) Encapsulation
(iv) Data Hiding

3. CLASS AS AN OBJECT FACTORY

A class can be viewed as an object factory, as it can create multiple objects.
E.g a class of cellphone can have numerous objects as Nokia, Samsung, Panasonic, etc. The features and functions of the objects are described within the class.

4. OBJECT IN JAVA

An object is an instance of a class.
E.g.-
(i) Nokia is an instance of class cellphone
(ii) Maruti is an instance of the class car.

5. OBJECTS ENCAPSULATE STATE (ATTRIBUTES) AND BEHAVIOUR (FUNCTIONS)

Objects are the building blocks of Object-Oriented Programming (OOP). They represent real-world entities with unique characteristics and functionalities :

5.1 Objects in Java Encapsulate

(i) State (Attributes) : These are the properties of the object, such as a Nokia cellphones memory capacity, number of tones, volume range, size, and shape.
(ii) Behaviour (Functions) : These define the actions the object can perform, such as sending and receiving calls, playing games, etc.

NODIA

The combination of these attributes and functions creates a unified entity, which is referred to as an Object. This encapsulation ensures that the object's state and behaviour are tightly bound, adhering to the principles of Object-Oriented Programming (OOP).

5.2 Message Passing/Function Calls Between Objects

In Object-Oriented Programming, objects interact and communicate with each other by message passing. This is a process where one object calls the predefined methods of another object to perform specific tasks.
For example, a Nokia cellphone object can make a call by executing a method (like dialNumber()) through a few key strokes. This interaction allows objects to work collaboratively, enabling modular and reusable code design.

6. CLASS AS A USER-DEFINED DATA TYPE

In Java, a class acts as a user-defined data type to represent objects. It provides a structure to define the attributes (data members) and behaviors (methods) of an object. Using a class, developers can create multiple instances (objects) with similar properties and functionality, ensuring modularity and reusability in code.

7. ACCESS SPECIFIER

Access specifiers in Java are used to control the visibility and accessibility of data members and member functions. There are three primary types of access specifiers:

(i) Public : Members declared as public are accessible from anywhere in the program, making them visible to the world.

(ii) Private : Members declared as private are accessible only within the class in which they are defined, ensuring data encapsulation and protection.

(iii) Protected : Members declared as protected are accessible within the same class, subclasses, and other classes in the same package.

8. JAVA - A STRONGLY TYPED LANGUAGE

Java is a strongly typed language where every variable and expression has a specific type. The compiler enforces type safety by checking that assigned values are compatible with the declared types, ensuring reliability and preventing type-related errors.

9. PRIMITIVE DATA TYPES

Integer, floating, character, boolean.

Data type	Description	Range of values	Size	Default value
byte	Byte length integer	-128 to +127 (signed)	1 byte = 8 bits	0
short	Short integer	-32,768 to 32,767	2 bytes = 16 bits	0
int	Integer	-2^{31} to $2^{31}-1$	4 bytes = 32 bits	0
long	Long integer	-2^{63} to $2^{63}-1$	8 bytes = 64 bits	0
float	Single precision floating no.	-3.4×10^{38} to 3.4×10^{38}	4 bytes = 32 bits	0.0f
double	Double precision floating no.	-1.7×10^{38} to 1.7×10^{38}	8 bytes = 64 bits	0.0d
char	Used to store characters.	Unicode is used to represent characters. (0 to 65,535)	2 bytes = 16 bits	'\ u0000'
boolean	Boolean values true or false.	true, false	1 bit	false

10. VARIABLES

A variable is a named memory location used to store a value, which can change during program execution, hence the term "variable." The value it holds must match the data type specified in its declaration.

Download INDIA's Best Study app NODIA

10.1 Rules for Naming Variables

Following proper naming conventions for variables ensures code readability and consistency.

(i) Start with a lowercase letter : Variable names should always begin with a lowercase letter to distinguish them from class names, which typically start with an uppercase letter. Examples include sum, avg, and count. This follows the standard camel-case naming convention used in Java.

(ii) Use camel-case for multi-word names : When a variable name consists of multiple words, the first word should start with a lowercase letter, and the first letter of each subsequent word should be capitalized. This is known as camel-case. For example, account-details and totals um clearly indicate their purpose while maintaining readability.

(iii) Use uppercase letters for constants : Constants, which are variables declared as final, should be named using all uppercase letters, with words separated by underscores for clarity. This convention distinguishes constants from regular variables. Examples include LIMIT, MAXIMUM_VALUE, and PI.

10.2 Declaration of Variables

The general format for declaring a variable in Java is:

```
data type variable Name;
```

For example:

```
int number;
```

In this example:

(i) number is declared as a variable of type int, meaning it can store integer values.

(ii) The declaration reserves 4 bytes of memory to store the integer value associated with number.

The data type specifies the kind of data the variable can hold, such as integers (int), floating-point numbers (float), characters (char), or boolean values (boolean). This ensures type safety and helps the compiler manage memory efficiently.

10.3 Representing Attributes by Variables

Attributes of an object can be represented using variables, which act as placeholders to store the corresponding values. These variables provide a way to describe the properties of an object.

Click Here to Buy book on Amazon

For example:

(i) The age of a student, which is a numerical property, can be represented using an integer variable.

```
int age;
```

Here, age is declared as an integer variable to store the age of the student.

(ii) Variables can also be initialized at the time of declaration. For instance:

```
int age = 14;
```

In this case, the variable age is declared and immediately assigned the value 14.

10.4 Types of Variable

Different types of variables are used in Java for various purposes, each serving a specific role in the program. They include:

(i) Local Variable : A variable declared within a method, constructor, or block is called a local variable. It is accessible only within the scope where it is defined and gets destroyed once the method or block finishes execution.

(ii) Instance Variable : These are non-static variables declared inside a class but outside any method, constructor, or block. They are tied to a specific object instance and are used to store object-level data.

(iii) Reference Variable : A variable that stores the reference (memory address) of an object rather than the object itself. For example:

```
Student student1 = new Student();
```

Here, student1 is a reference variable that holds the reference to the Student object.

(iv) Static Variable : A variable declared with the static keyword is shared among all instances of a class. It belongs to the class rather than any individual object, making it useful for shared properties or constants.

(v) Final Variable : A variable declared with the final keyword becomes a constant, meaning its value cannot be changed once assigned. Final variables are often used for fixed values such as PI.

11. OBJECTS AS INSTANCES OF A CLASS

In Java, an object is an instance of a class, created to represent and manipulate data defined by that class. Similar to variables, an object is assigned a name that is used to reference it. However, unlike primitive variables, the data type of an object is the class it belongs to.

```
class Student {
    int age;
    String name;
}

// Creating an object of the Student
class
Student student1 = new Student();
```

Here

(i) Student is the class name, which also serves as the data type for the object.

(ii) student1 is the object (or instance) of the Student class.

(iii) The object student1 can now access the attributes (age, name) and methods defined in the Student class.

12. COMPOSITE DATA TYPE

A composite data type is a data type formed by combining one or more primitive data types. It allows for the creation of complex structures that can store multiple related values in a single entity. In Java, a class is an example of a composite data type, as it can group different types of data together.

Example of class as a composite data type :

```
class student {
  int rollNo;
  String name, address;
}
```

13. VARIABLE DECLARATION FOR COMPOSITE TYPE

In Java, variables for composite data types (classes) are declared and initialized in a structured way. Declaring an object is a two-step process:

Syntax :

```
ClassName variable;
variable = new ClassName();
```

Steps for Declaring an Object:

(i) Declaring object (Step 1) : A variable is declared to refer to an object of a specific class. At this stage, no memory is allocated for the object; the variable simply holds a reference to the object.

```
ClassName objectName;
```

Example -

```
  Sum ob;
```

Here, ob is a reference variable for an object of the Sum class. However, it does not yet point to any actual object.

NODIA

Download INDIA's Best Study app NODIA

(ii) Instantiation (Step 2) : Memory is allocated at runtime for the object using the new keyword, and the reference variable is linked to this newly created object.

```
objectName = new ClassName();
```

Example

```
ob = new Sum();
```

At this point, memory is allocated for the object ob, and it can now be used to access the attributes and methods of the Sum class.

14. NEW OPERATOR

The new operator in Java is used to allocate memory for an object at runtime. It creates an instance of a class and returns a reference to that object. To understand the functioning of the operator new, consider the following example :

```
class Sum {
  int a, b;
  public static void main(String a[])
{
    Sum obj;
    obj = new Sum();
  }
}
```

Statement : Sum obj;
Description : NULL (obj)
Statement : obj = new Sum();
Description : obj now contains the address of the space allocated for the object by the new operator. new allocates memory to an object at runtime.

MULTIPLE CHOICE QUESTION

1. Which of the following is NOT a feature of Object-Oriented Programming (OOP) implemented by classes in Java?

(a) Abstraction
(b) Inheritance
(c) Encapsulation
(d) Compilation

Ans : SQP 2024

(d) Compilation
Compilation is not a feature of OOP. focuses on abstraction, inheritance, encapsulation and polymorphism.

Download INDIA's Best Study app NODIA

2. What is the primary purpose of the new operator in Java?

(a) To declare a variable
(b) To allocate memory for an object at runtime
(c) To define a class
(d) To initialize a constant

Ans : MAIN 2010

(b) To allocate memory for an object at runtime
The new operator dynamically allocates memory for an object during runtime.

3. Which of the following correctly declares and initializes an object in Java?

(a) Student student1;
(b) Student1 = Student();
(c) Student student1 = new Student();
(d) New Student = student1();

Ans : COMP 2003

(c) Student student1 = new Student();
This syntax correctly declares an object (student1) and initializes it with a new instance of the Student class.

NODIA

4. Which access specifier allows members to be accessible within the same class, subclasses, and other classes in the same package?

(a) Private
(b) Public
(c) Protected
(d) Default

Ans : SQP 2014

(c) Protected
The protected access modifier permits access within the same package and subclasses.

5. In Java, what is the default value of a boolean variable?

(a) Null
(b) 0
(c) True
(d) False

Ans : MAIN 2006

(d) False
The default value of a boolean variable in Java is false.

Download INDIA's Best Study app NODIA

6. Which type of variable is declared within a method and is destroyed once the method is finished?

(a) Instance Variable

(b) Local Variable

(c) Static Variable

(d) Final Variable

Ans : COMP 2007

(b) Local Variable
Local variables are declared inside methods and exist only during the method's execution.

7. What is the size of an int data type in Java?

(a) 1 byte

(b) 2 bytes

(c) 4 bytes

(d) 8 bytes

Ans : SQP 2018

(c) 4 bytes
In Java, the int data type occupies 4 bytes of memory (32 bits).

8. What is the primary feature of a final variable in Java?

(a) It can store the reference of multiple objects.

(b) Its value can be modified during runtime.

(c) Its value cannot be changed once assigned.

(d) It is accessible only within the class.

Ans : MAIN 2009

(c) Its value cannot be changed once assigned.
The final keyword ensures that a variable's value remains constant once initialized.

9. What does the following line of code do?

```
Student student1 = new Student();
```

(a) Declares a class named Student.

(b) Creates a new object of the Student class and assigns its reference to student1.

(c) Declares a local variable named Student.

(d) Assigns a new class to the variable student1.

Ans : COMP 2011

(b) Creates a new object of the Student class and assigns its reference to student1.
This statement creates a new instance of the Student class and stores its reference in student1.

10. What is the primary purpose of encapsulation in Java?

(a) To enable inheritance between classes

(b) To hide the implementation details and protect data

(c) To provide faster compilation

(d) To improve runtime efficiency

Ans : SQP 2015

(b) To hide the implementation details and protect data
Encapsulation restricts direct access to data and methods, ensuring controlled access.

11. A class is called :

(a) Object Factory

(b) User defined data type

(c) Composite data type

(d) All of these

Ans : MAIN 2020

(d) All of these
A class can be referred to as an object factory, a user-defined data type, or a composite data type.

12. Which of the following is not a keyword?

(a) Void

(b) Byte

(c) Object

(d) Public

Ans : COMP 2023

(c) Object
The word object is not a reserved keyword in Java.

13. A variable whose value is not associated to an object and its value exists throughout the life of the program is a?

(a) Local variable

(b) Static variable

(c) Final variable

(d) None of these

Ans : SQP 2024

(b) Static variable
Static variables are associated with the class, not any specific object, and persist throughout the program.

14. Which of the following operator is used to allocate memory to an object?

Downloaded from www.pdf.tube

(a) Dot
(b) New
(c) Both (a) and (b)
(d) None of these

Ans : MAIN 2012

(b) New
The new operator dynamically allocates memory for objects.

15. The characteristics of a class are represented by which of the following variable?
(a) Local variables
(b) Instance variables
(c) Static variables
(d) Final variables

Ans : COMP 2014

(b) Instance variables
Instance variables define the attributes or properties of an object.

16. In the below statement, what is the name of the object being created?
Student stu = new Student();
(a) Student
(b) stu
(c) Stu
(d) None of these

Ans : COMP 2018

(b) stu
The object being created is named stu.

17. In the below class, choose the name of the data members :

```
class Test {
    String name;
    int age;
    public static void main() {
        float x = 5;
        int p;
    }
}
```

(a) name, age
(b) x, p
(c) name, x
(d) age, p

Ans : SQP 2019

(a) name, age
Data members are variables defined within the class but outside of methods (name and age).

NODIA

18. For which of the following data types, the size of the variable declared will be fixed?
(a) Primitive
(b) Reference
(c) Composite
(d) None of these

Ans : MAIN 2020

(a) Primitive
Primitive data types in Java have fixed sizes (e.g., int is 4 bytes).

19. A class implements which of the following Object Oriented programming characteristic?
(a) Encapsulation
(b) Inheritance
(c) Abstraction
(d) Polymorphism

Ans : COMP 2022

(a) Encapsulation
A class encapsulates data and behavior within a single unit.

20. The messages are passed in a class using :
(a) Local variables
(b) Function parameters
(c) Objects
(d) None of these

Ans : SQP 2016

(b) Function parameters
Messages (data) are passed between methods via function parameters.

21. **Statement 1 :** In Java, the private access specifier ensures that members can only be accessed within the same class.
Statement 2 : The protected access specifier allows access to members only within the same class.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : MAIN 2011

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : The private access specifier restricts the visibility of class members so they can only

be accessed within the class itself, ensuring data encapsulation.
Statement 2 : The protected access specifier allows access within the same class, subclasses, and other classes within the same package, not just within the class itself.

22. **Statement 1 :** The new operator in Java is used to create objects dynamically at runtime.
Statement 2 : The new operator is also responsible for deleting objects from memory when they are no longer used.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : COMP 2015

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : The new operator dynamically allocates memory for objects during runtime.
Statement 2 : Java uses garbage collection to automatically manage memory and delete objects that are no longer used. The new operator does not handle object deletion.

23. **Statement 1 :** An object is an instance of a class and can have its own unique state and behavior.
Statement 2 : A class can only have one object created from it.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : SQP 2019

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : Objects are specific instances of a class, encapsulating state (attributes) and behavior (methods).
Statement 2 : A class can serve as a blueprint for creating multiple objects, each with its own unique state and behavior.

24. **Statement 1 :** In Java, local variables are destroyed once the method or block in which they are defined finishes execution.
Statement 2 : Instance variables are tied to specific objects and their lifetime depends on the object.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : MAIN 2018

(a) Both statements are true.
Statement 1 : Local variables are scoped to the method or block in which they are defined and are destroyed when the block finishes execution.
Statement 2 : Instance variables belong to objects, and their lifetime is tied to the object's existence.

25. **Statement 1 :** The public access specifier makes class members accessible from anywhere in the program.
Statement 2 : The private access specifier makes class members accessible from subclasses of the class.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : COMP 2019

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : The public access specifier makes members visible to all classes in the program.
Statement 2 : The private access specifier restricts visibility to the class itself, not its subclasses.

26. **Statement 1 :** A class is a blueprint for creating objects.
Statement 2 : A variable of a primitive type can also act as an object.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : SQP 2020

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : A class provides the structure for creating objects with shared attributes and methods.
Statement 2 : Primitive types (e.g., int, boolean) are not objects in Java; they are basic data types.

27. **Statement 1 :** In Java, variables declared with the final keyword cannot be reassigned after their initial assignment.
Statement 2 : The final keyword can also be used to prevent a class from being subclasses.
(a) Both statements are true.
(b) Both statements are false.

Click Here to Buy book on Amazon

(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : MAIN 2021

(a) Both statements are true.
Statement 1 : The final keyword ensures a variable cannot be reassigned, making it a constant.
Statement 2 : The final keyword applied to a class prevents it from being extended.

28. **Statement 1 :** An object's state is defined by its attributes, while its behavior is defined by its methods.
Statement 2 : An object in Java can exist without being associated with a class.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : COMP 2024

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : An object's state is represented by its attributes, and its behavior is defined by its methods.
Statement 2 : An object must always be created from a class, which serves as its blueprint.

29. **Statement 1 :** Composite data types in Java can combine different types of data into a single entity.
Statement 2 : Primitive data types like int and float are examples of composite data types.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : SQP 2023

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : Composite data types, such as classes, group multiple related values into a single structure.
Statement 2 : Primitive types like int and float are basic data types, not composite types.

30. **Statement 1 :** Java is a strongly typed language, ensuring all variables have specific types.
Statement 2 : The type of a variable in Java can change during program execution.
(a) Both statements are true.
(b) Both statements are false.
(c) Statement 1 is true, but Statement 2 is false.
(d) Statement 1 is false, but Statement 2 is true.

Ans : COMP 2022

(c) Statement 1 is true, but Statement 2 is false.
Statement 1 : Java enforces strong typing, meaning every variable must be declared with a specific type, which ensures type safety.
Statement 2 : A variable's type cannot change during execution; it must match the type declared at compile time.

31. A student declares an integer variable z and wants to assign a value of 15 to it. Which of the following statements is correct?
A. int z = 15;
B. z = new int(15);
C. int z = new 15;
D. z int = 15;
(a) Only A
(b) Both A and B
(c) Only C
(d) Both B and C

Ans : MAIN 2017

(a) Only A
int z = 15; is the correct way to declare and assign a value to an integer.

32. Which of the following correctly represents the declaration and memory allocation for an object of class Student?
A. Student obj;
B. obj = new Student();
C. Student obj = new Student();
D. new Student obj;
(a) Only C
(b) Only B
(c) Both A and B
(d) Both B and C

Ans : MAIN 2000

(d) Both B and C.
Declaration (Student obj;) and initialization (obj = new Student();) can be combined or performed separately.

33. The following code snippet is executed by a student :

```
int x = 10;
x += 5;
```

What will be the value of x after the execution of the code?
A. 5
B. 10
C. 15

NODIA

D. 20

(a) Only A

(b) Only C

(c) Both B and C

(d) Both C and D

Ans : COMP 2005

(b) Only C

x += 5 increments x by 5, so the value becomes 15.

34. Which of the following statements about the new operator in Java is correct?

A. It allocates memory for a new object at runtime.

B. It initializes all the instance variables of the object.

C. It is used to declare a variable of a class.

D. It is a reserved keyword in Java.

(a) Only A

(b) Only D

(c) Both A and B

(d) Both A and D

Ans : SQP 2008

(d) Both A and D.

The new operator allocates memory for an object and is a reserved keyword.

35. Which of the following correctly represents an instance variable in Java?

A. A variable declared inside a method.

B. A variable declared within a class but outside any method.

C. A variable declared with the keyword static.

D. A variable declared as a constant with final.

(a) Only A

(b) Only B

(c) Both A and C

(d) Both B and C

Ans : MAIN 2005

(b) Only B

Instance variables are declared within a class but outside any method.

36. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to calculate the factorial of a number.

```
int factorial(int n)
{
  for (int i = 1; i <= n; i++) → (1)
    int fact = 1; → (2)
    fact = fact * i; → (3)
    return fact; → (4)
}
```

(a) (2) (1) (3) (4)

(b) (1) (3) (2) (4)

(c) (3) (2) (1) (4)

(d) (4) (2) (1) (3)

Ans : SQP 2004

(a) (2) (1) (3) (4)

Variables are initialized first, followed by iteration and calculation.

37. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to check whether a number is even or odd.

```
int factorial(int n)
{
   for (int i = 1; i <= n; i++) → (1)
     int fact = 1; → (2)
     fact = fact * i; → (3)
     return fact; → (4)
}
```

(a) (1) (2) (3) (4) (b) (3) (4) (1) (2)

(c) (1) (3) (2) (4) (d) (2) (1) (4) (3)

Ans : COMP 2009

(a) (1) (2) (3) (4)

The correct sequence involves iteration, initialization, condition check and result.

38. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to reverse a string.

```
String reverse(String str)
{
    char[ ] arr = str.toCharArray();
→ (1)
    String reversed = "”; → (2)
    for (int i = arr.length - 1; i >=
0; i--) → (3)
    reversed += arr[i]; → (4)
    return reversed; → (5)
}
```

(a) (1) (2) (3) (4) (5)

(b) (2) (1) (3) (4) (5)

(c) (1) (3) (2) (4) (5)

(d) (1) (2) (4) (3) (5)

Ans : MAIN 2005

Downloaded from www.pdf.tube

(b) (2) (1) (3) (4) (5)
The reversed string is built step-by-step by iterating through the character array.

39. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to find the greatest of two numbers.

```
void findGreatest(int a, int b)
{
    if (a > b) → (1)
      System.out.println(a + " is greater"); → (2)
    else → (3)
       System.out.println(b + " is greater"); → (4)
}
```

(a) (1) (3) (2) (4)
(b) (1) (2) (3) (4)
(c) (3) (4) (1) (2)
(d) (2) (1) (3) (4)

Ans : SQP 2006

(b) (1) (2) (3) (4)
Condition checking is followed by printing the respective results.

40. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to find the sum of all elements in an array.

```
int calculateSum(int[] arr)
{
    int sum = 0; → (1)
    for (int i = 0; i < arr.length; i++) → (2)
    sum += arr[i]; → (3)
    return sum; → (4)
}
```

(a) (1) (2) (3) (4)
(b) (2) (1) (3) (4)
(c) (3) (2) (1) (4)
(d) (1) (3) (2) (4)

Ans : COMP 2011

(a) (1) (2) (3) (4)
Initialize sum, iterate through the array, add elements, and return the sum.

Download INDIA's Best Study app NODIA

ASSERTION AND REASON

NODIA

41. **Assertion (A) :** The new operator in Java allocates memory to an object at runtime.
Reason (R) : The new operator in Java initializes all instance variables to their default values.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2017

(a) Both A and R are true, and R is the correct explanation of A.
The new operator allocates memory at runtime and initializes all instance variables of the object to their default values (e.g., 0 for integers, null for objects).

42. **Assertion (A) :** A static variable in Java exists throughout the life of the program.
Reason (R) : Static variables in Java are created and initialized each time a method is called.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : SQP 2011

(c) A is true, but R is false.
Static variables exist throughout the life of the program but are created only once when the class is loaded, not each time a method is called.

43. **Assertion (A) :** An object is an instance of a class.
Reason (R) : Objects encapsulate the attributes and behavior of a class into a single unit.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2012

(a) Both A and R are true, and R is the correct explanation of A.

Objects are instances of a class, and they encapsulate both data (attributes) and methods (behavior).

44. **Assertion (A) :** Local variables in Java can be accessed outside the method where they are declared.
Reason (R) : Local variables in Java are only alive during the execution of the method in which they are declared.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : SQP 2010

(d) A is false, but R is true.
Local variables cannot be accessed outside their declaring method, but they exist only during the method's execution.

45. **Assertion (A) :** Access specifiers in Java are used to control the visibility of data members and member functions.
Reason (R) : Protected access specifier in Java makes members visible only within the same class.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2015

(c) A is true, but R is false.
Access specifiers control visibility, but protected members are visible to the same class, subclasses, and other classes in the same package.

46. **Assertion (A) :** A class in Java can be used as a composite data type.
Reason (R) : A class can encapsulate both attributes and methods to represent objects.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2005

(a) Both A and R are true, and R is the correct explanation of A.
Classes encapsulate data and methods, making them composite data types.

47. **Assertion (A) :** Instance variables in Java are accessible without creating an object of the class.
Reason (R) : Instance variables are associated with objects of a class, and memory is allocated to them when the object is created.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2013

(d) A is false, but R is true.
Instance variables require an object to be accessed, but they are allocated memory only when the object is created.

48. **Assertion (A) :** Static variables are shared among all objects of a class.
Reason (R) : A static variable is created only once and is stored in the heap memory.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2018

(c) A is true, but R is false.
Static variables are shared, but they are stored in the method area (class memory), not in the heap.

49. **Assertion (A) :** In Java, a variable name can start with a number.
Reason (R) : Java has strict rules for naming variables to avoid ambiguity during compilation.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2010

(d) A is false, but R is true.

Click Here to Buy book on Amazon

Variable names in Java cannot start with a number, but Java does enforce strict naming rules for clarity and to avoid errors.

50. **Assertion (A) :** Primitive data types in Java include int, float, char, and String.
Reason (R) : String in Java is a class, not a primitive data type.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2019

(d) A is false, but R is true.
String is a class in Java, whereas primitive data types include int, float, char, boolean, etc.

TWO MARK QUESTIONS

51. What does a class encapsulate?

Ans : MAIN 2024

A class encapsulates:
(i) Data (Attributes) : Properties of the objects it creates, like color or speed in a Car class.
(ii) Behavior (Methods) : Actions the objects can perform, such as accelerate() or brake().
This ensures objects maintain their state and behavior, adhering to OOP principles like abstraction and encapsulation.

52. What are the different set of data types available in Java?

Ans : COMP 2023

Java provides two main categories of data types:
(i) Primitive Data Types : Basic data types like int, float, char, and boolean.
(ii) Reference Data Types : Includes objects, arrays, and classes, which refer to memory locations storing complex data structures.

53. Difference between public and private modifiers for members of a class.

Ans : SQP 2022

(i) Public members : Can be accessed in any class contained in any package.
(ii) Private members : Can only be accessed within the declared class only.

54. List various reference data types of Java.

Ans : MAIN 2021

Various Reference Data Types in Java are as follows :
(i) Class : Defines the blueprint for objects and is a reference data type.
(ii) Object : A variable of type class that holds the reference to the instance of a class.
(iii) Array : A collection of elements of the same type, stored in contiguous memory locations.
(iv) Interface : Defines a contract for classes to implement, serving as a reference data type.

55. Write a Java statement to compute discount by the following method :
Divide the variable total by 100. Multiply the result by MP.

Ans : SQP 2001

double x = (double) total / 100 * MP;

56. What is the difference between an object and a class?

Ans : COMP 2003

Class : Class encapsulates the characteristics and functions that can be applied on objects. It is an object factory.
Object : Object is an instance of the class. Every object has a unique identity determined by its characteristics.
Example:Student is a class. Naman, Amit, Suman are objects of type Student.

57. (i) Why is a class called as factory of objects?
(ii) Why is a class known as composite data type?

Ans : MAIN 2004

(i) A class describes the features and functions of the objects that can be created from it. It can be used to create any number of objects, so it is called a factory of objects.
(ii) A class is called a composite data type because it binds up (encapsulates) one or more primitive types together as a single data type.E.g.

```
class Abc {
      int x;
      float y;
      }
```

Downloaded from www.pdf.tube

58. (i) Mention any two attributes required for class declaration.
(ii) Explain instance variable. Give an example.

Ans : SQP 2005

(i) Attributes of a class can be primitive type or reference type. To complete the class, we need to specify data members, constructor, and member functions.

(ii) An instance variable is a variable which is associated with the object of a class. It is also known as a data member or member variable. Its scope is the entire class in which it is defined. The scope of these variables can be further enhanced or restricted based on the access specifiers private, protected, public, or default. Example :

```
class Student {
    int rollNumber;
    String name;
    }
```

59. (i) Define Instance Variable. Give an example of the same.
(ii) Assign the value of pie (i.e. 3.142) to a variable with requisite data type.

Ans : COMP 2005

(i) An instance variable is a member of the class which has its value associated with the objects of the class. Example :

```
class Student {
int roll;
  String name;
  Student(int a, String b) {
        roll = a;
        name = b;
        }
  }
```

(ii)

```
double p = Math.PI;
```

60. What is a class in Java?

Ans : SQP 2009

A class in Java is a blueprint or prototype that defines a group of objects with similar properties and behaviours. It encapsulates data (attributes) and functions (methods) into a single unit, which is known as encapsulation. This structure allows developers to model real-world entities in a program. A class also supports abstraction, enabling the hiding of unnecessary details while exposing essential features.

61. What are the features of Object-Oriented Programming implemented by classes?

Ans : MAIN 2005

Classes in Java implement the following OOP features :

(i) Abstraction : Simplifying complex systems by modeling classes appropriate to the problem.
(ii) Inheritance : Enabling new classes to reuse the properties and methods of existing ones.
(iii) Encapsulation : Bundling data and methods into a single unit, ensuring controlled access.
(iv) Data Hiding : Restricting direct access to certain members of a class to protect the data and maintain integrity.

62. What does an object encapsulate in Java?

Ans : COMP 2011

An object in Java encapsulates:
(i) State (Attributes) : These are the properties or data that define the object's characteristics. For example, a Car object might have attributes like color, brand, and speed.
(ii) Behavior (Methods) : These define the actions the object can perform. For instance, a Car object might have methods like accelerate() or brake().

By encapsulating state and behavior, objects form the foundation of Object-Oriented Programming (OOP). This ensures that data and functions are tightly bound, promoting modularity and re-usability.

63. What are the primary access specifiers in Java?

Ans : SQP 2016

Access specifiers control the visibility and accessibility of members in a class:
(i) Public : Members declared as public are accessible from any class in the program, making them globally visible.
(ii) Private : Members declared as private are accessible only within the class where they are defined, ensuring encapsulation and data protection.
(iii) Protected : Members declared as protected are accessible within the same class, subclasses, and other classes in the same package. This strikes a balance between accessibility and encapsulation.

64. Differentiate between primitive and reference data types in Java.

NODIA

Ans : COMP 2013

Aspect	Primitive Data Types	Reference Data Types
Definition	Basic data types provided by Java.	Include objects, arrays, and classes.
Storage	Store simple values directly in memory.	Store references (memory addresses) pointing to the actual data.
Size	Have a fixed size (e.g., int is 4 bytes).	Size depends on the type and size of the object or structure.
Examples	int, float, char, boolean, etc.	Objects (e.g., Student), arrays, and user-defined classes.

65. Explain the steps for declaring and initializing an object in Java.

Ans : SQP 2012

Declaring and initializing an object in Java is a two-step process:

(i) **Declaration :** A reference variable is declared to refer to the object. For example:

```
ClassName objectName;
```

Here, objectName is declared but does not yet point to any object.

(ii) **Instantiation :** Memory is allocated for the object using the new operator, and the reference variable is linked to the object. Example:

```
objectName = new ClassName();
```

66. A student is trying to convert a string representing a decimal number into a float value so that they can multiply it by a constant factor. However, the code contains an error. Name the error (syntax/logical/runtime). Correct the code so that it compiles and runs correctly.

```
String numStr = "5.5";

int num = Float.parseFloat(numStr);

float result = num * 2;

System.out.println("Result : " + result);
```

Ans : SQP 2004

```
String numStr = "5.5";

float num = Float.parseFloat(numStr);

float result = num * 2;

System.out.println("Result : " + result);
```

67. What is the role of the new operator in Java?

Ans : SQP 2015

The new operator in Java is used to allocate memory for an object at runtime. It creates an instance of a class and returns a reference to the memory location where the object is stored. For example:

```
Student obj = new Student();
```

Here, new creates an object of the Student class and assigns its reference to the variable obj. This ensures that memory is dynamically allocated only when needed, supporting efficient resource management.

68. A student is trying to parse a string into an integer and add a value to it, but the code contains an error. Identify the type of error (syntax/logical/runtime) and correct the code.

```
String numStr = "25.0";
int num = Integer.parseInt(numStr);

int result = num + 10;

System.out.println("Result : " + result);
```

Ans : COMP 2005

Error Type : Runtime Error

```
String numStr = "25.0";
float num = Float.parseFloat(numStr);

float result = num + 10;

System.out.println("Result : " + result);
```

69. A student is attempting to convert a string into a double value and divide it by a constant. However, the code contains an error. Identify the type of error (syntax/logical/runtime) and correct the code.

Click Here to Buy book on Amazon

```
String numStr = "12.34";
int num = Double.parseDouble(numStr);

double result = num / 3;

System.out.println("Result : " +
result);
```

Ans : SQP 2010

```
Error Type : Syntax Error
String numStr = "12.34";
double num = Double.
parseDouble(numStr);

double result = num / 3;

System.out.println("Result : " +
result);
```

70. A student is trying to convert a string into a boolean value but encounters an issue. Identify the type of error (syntax/logical/runtime) and correct the code.

```
String boolStr = "True";
boolean isTrue = Boolean.
parseBoolean(boolStr);

if (isTrue == "true") {
    System.out.println("The value is
true.");
} else {
    System.out.println("The value is
false.");
}
```

Ans : COMP 2006

Error Type : Logical Error

```
String boolStr = "True";
boolean isTrue = Boolean.
parseBoolean(boolStr);

if (isTrue) {
    System.out.println("The value is
true.");
} else {
    System.out.println("The value is
false.");
}
```

Download INDIA's Best Study app NODIA

71. A student is trying to parse a string containing an integer and then add a value to it. However, the code has an error. Name the error (syntax/logical/runtime). Correct the code so that it compiles and runs correctly.

```
String value = "42";

double number = Integer.
parseInt(value);

int total = number + 10;

System.out.println("Total : " +
total);
```

Ans : MAIN 2010

```
String value = "42";

int number = Integer.parseInt(value);

int total = number + 10;

System.out.println("Total : " +
total);
```

NODIA

72. A student is trying to find the length of a string after converting it to uppercase. However, the code contains an error. Name the error (syntax/logical/runtime). Correct the code so that it compiles and runs correctly.

```
String text = "Hello";

int length = text.toUppercase().
length();

System.out.println("Length : " +
length);
```

Ans : SQP 2011

```
String text = "Hello";

int length = text.toUpperCase().
length(); // Correct method name

System.out.println("Length : " +
length);
```

73. A student is trying to convert a string into an integer and calculate its square. However, the code contains an error. Name the error (syntax/logical/runtime). Correct the code so that it compiles and runs correctly.

```
String numStr = "10";
square = Math.pow(Integer.
parseInt(numStr), 2);

System.out.println("Square : " +
square);
```

Ans : COMP 2012

```
String numStr = "10";

int square = (int) Math.pow(Integer.
parseInt(numStr), 2);

System.out.println("Square : " +
square);
```

14. Explain the concept of a class as an object factory.

Ans : MAIN 2013

A class in Java can be thought of as an object factory because it acts as a template to create multiple objects. Each object created from the class shares the same structure and behavior defined by the class, but with its own unique data. For instance, a Cellphone class can create objects like Nokia, Samsung, and Panasonic. Each object represents a real-world cellphone but adheres to the common features and functions described within the class.

15. A student is trying to parse a string to a double and find its square root. However, the code has an error. Name the error (syntax/logical/runtime). Correct the code so that it compiles and runs correctly.

```
String number = "81";

int value = Double.
parseDouble(number);

double root = Math.sqrt(value);

System.out.println("Square Root : "
+ root);
```

Ans : COMP 2014

```
String number = "81";

double value = Double.
parseDouble(number);

double root = Math.sqrt(value);

System.out.println("Square Root : "
+ root);
```

16. Observe the code snippet given below, state the name of the

```
class myClass {
    static int x = 7;
    int y = 2;
     public static void main(String
args[]) {
        myClass obj = new myClass();
        System.out.println(x);
        obj.sampleMethod(5);
        int a = 6;
        System.out.println(a);
    }
    void sampleMethod(int n) {
        System.out.println(n);
        System.out.println(y);
    }
}
```

(i) Method argument or argument variable.
(ii) Class variable
(iii) Local variable
(iv) Instance variable

Ans : SQP 2015

(i) Argument variable = n
(ii) Class variable = x
(iii) Local variables = obj, a
(iv) Instance variable = y

FIFTEEN MARKS QUESTIONS

17. Prepare a list of attributes (data values), methods, and their description that you would expect the objects of the following classes to handle :
(i) Class Employee
(ii) Class Student
(iii) Class Bank Customer

Ans : MAIN 2024

(i) Class Employee

Attributes	Description	Methods	Description
Employee number	Unique number or code that identifies an employee	Read_data	This method reads the employee data and writes it in a file.
Emp_name	Name of the employee	Disp_ emp_list()	This method prints the list of all employees.
Address	The address of the employee	Add_ emp()	This method adds details of new employees in the file.

Downloaded from www.pdf.tube

Phone_ number	Contact number of the employee	Delete_ emp()	If an employee leaves, their information is erased from the file.
Qualifications	The academic and technical qualifications of the employee	Modify_ emp()	If the information related to an employee is changed, the change is made in the file.
Previous experience	The work experience of the employee before joining the company	Search_ emp()	Searching an employee on the basis of their employee number and printing their details.
Date_of_ joining	Joining date of the employee in month, day, and year	Calculate_ net_ salary()	This method calculates the net salary of the employee.

(ii) Class Student

Attributes	**Description**	**Methods**	**Description**
Admn_ number	Admission number	Getdata()	This method inputs the data values.
Date_of_ admn	Admission date of the student	Showdata()	This method displays all the details.
Roll_ number	Roll number of the student	Compute_ total()	This method computes the total marks scored by the student.
Name	Student's name	Compute_ percent()	This method calculates the percentage of the student.
Class	The class to which the student currently belongs	Assign_ grade()	It assigns the grades to students depending on their percentages.
Section	The class section to which the student belongs	Ascending_ roll()	Arranges the list in ascending order of the roll numbers.

NODIA

Address	**Student's residential address**	**Descending_ rank()**	**Arranges the name list in descending order of the percentage.**
Phone_ number	Contact number of the student	-	-
Date_of_ birth	Birth date in month, day, and year	-	-
Eng_ marks	English marks of the student	-	-
Maths_ marks	Student's marks in math	-	-
Sci_marks	Student's science marks	-	-
SST_ marks	Student's SST marks	-	-
Comp_ marks	Student's computer marks	-	

(iii) Class Bank Customer

Attributes	**Description**	**Methods**	**Description**
Cust_id	The customer's identification code	Get_data()	This method takes the inputs from the user.
Cust_name	The customer's name	Show_data()	This method displays the customer data.
Address	The customer's address	Deposit_amt()	Calculates the balance when the amount is deposited in the account.
Account_ num	The customer account number	Withdrawl()	Calculates the balance when the amount is withdrawn from the account.
Balance	The balance amount in the account	-	-

78. Prepare a list of classes, the attributes, and the methods that you would expect the following real-world systems to handle.
(i) Staff management system of a company
(ii) A banking system
(iii) Railway reservation system

Ans : SQP 2017

(i) Staff management system of a company

Class 1 : Personal Details
Objects : All staff members
Attributes : Code, name, address, contact number, marital status, hobbies, passport number, email ID, academic qualifications, technical qualifications, extracurricular report, previous work experience, trainings undergone.
Methods : Get data, show data, search an employee based on the name, arrange the list of names alphabetically, arrange the list of names in descending order of their number of years of experience.
Class 2 : Salary Details
Objects : All staff members
Attributes : Code, name, basic salary, DA, HRA, PF, travel allowance.
Methods : Get data, show data, calculate net salary, display the list of names in descending order of salary, print the salary slip (pay slip).

(ii) A banking system

Class 1 : Personal Details
Objects : All customers of the bank
Attributes : Code, name, address, contact number, amount of tax paid, account number, name of the nominee.
Methods : Get data, show data, search a customer based on the account number, arrange the list of names alphabetically.
Class 2 : Account Details
Objects : All bank customers
Attributes : Code, account number, name of account holder, balance amount, type of account, type of latest transaction (D or W for deposit or withdrawal), cheque book number, amount of latest transaction.
Methods : Get data, show data, modify the balance on deposit, modify the balance on withdrawal.
Class 3 : Staff Personal Details
Objects, attributes, methods : Same as that of part (i) Staff management.
Class 4 : Payslip
Objects, attributes, methods : Same as that of Salary Details of part (i) Staff management.

(iii) Railway reservation system

Class 1 : Train
Objects : All trains
Attributes :
Train number, train name, first station, last station, list of intermediate stations, time of departure from the starting point, days on which the train is available, list of times of arrival at all intermediate stations, classes available.
Methods :
Get data, show data, calculate total time it takes to reach a station.
Class 2 : Passenger
Objects : All passengers
Attributes :
Passenger name, age, date of travel, train number, train name, station from where to board the train, station at which to get down.
Methods :
Calculate the ticket amount, allot seat number.
Class 3 : Staff Details
Objects, attributes, methods : Same as that of Personal Details of part (i) Staff management.
Class 4 : Salary Details
Objects, attributes, methods : Same as that of Salary Details of part (i) Staff management.

79. Explain how the objects communicate with each other. Illustrate the concept by taking an example of the objects around you.

Ans : COMP 2018

Take an example of the student objects interacting with each other to exchange information about the percentage. One student, say A, passes their English, Maths, Hindi, Science, Computer, and Social Science marks to another student. The other student, say B, performs a method to calculate the total and percentage and tells the result to A. This involves two objects, A and B. The values passed to another object, B, are called arguments of the message. The process is performed by B using the percentage calculation method, and the result is returned by B to A. Hence, objects communicate with each other through the message-passing mechanism. This is possible because all objects of a class share common attributes and methods. These methods can be called at times when required. So, message

Click Here to Buy book on Amazon

passing can be termed as method calling by passing arguments.

80. Write a Java class Sum that initializes the two integer numbers 10 and 25 to variables and prints the result after :
(i) Addition
(ii) Subtraction
(iii) Multiplication
(iv) Division

Ans : MAIN 2019

```
class Sum {
    public static void main() {
          int num1 = 10, num2 = 25,
sum, diff;
        int product;
        float quotient;

        sum = num1 + num2;
        diff = num1 - num2;
        product = num1 * num2;
       quotient = (float) num1 / num2;

         System.out.println("Sum = "
+ sum);
       System.out.println("Difference
= " + diff);
         System.out.println("Product
= " + product);
        System.out.println("Quotient
= " + quotient);
    }
}
```

Variables used :
num1, num2 : Store the two numbers to be operated.
sum : Stores the added result of num1 and num2.
diff : Stores the subtracted result of num1 and num2.
product : Stores the multiplied result.
quotient : Stores the quotient after dividing num1 by num2.

81. Write a Java program that prints the computed kinetic energy by the formula :

$$\text{K.E.} = \frac{1}{2}mv^2, \text{where } m = 4.5\text{kg}, v = 10\text{m/s}.$$

Ans : SQP 2020

```
class Kinetic {
    public static void main() {
        double m = 4.5, v = 10.0, E;
        E = 0.5 * m * v * v;
         System.out.println("Kinetic
Energy = " + E);
    }
}
```

Variables used :
m : mass of body in kg.
v : velocity of moving body in m/s.
E : kinetic energy.

82. Write a Java program that computes the double and half of a, where a stores 10.92.

Ans : MAIN 2021

```
class Compute {
    public static void main() {
        double a, d, h;
        a = 10.92;
        d = 2 * a;
        h = 1 / 2.0 * a;
       System.out.println("Double of
" + a + " = " + d);
         System.out.println("Half of
" + a + " = " + h);
    }
}
```

Variables used :
a : the given number
d : stores the double of a
h : stores the half of a

83. Given that the cost price of an article is 60.25 and is sold for ₹ 89.75. Write a program that calculates the profit percent of the sale using the formula :

$$\text{Profit in } \% = \frac{\text{Profit}}{\text{C.P.}} \times 100$$

Where profit = S.P. - C.P.

Ans : COMP 2022

```
class Profit {
    public static void main() {
       double cp = 60.25, sp = 89.75;
        double pr, percentProfit;
        pr = sp - cp;
       percentProfit = pr / cp * 100;
        System.out.println("Profit in
per cent = " + percentProfit + "%");
    }
}
```

Variables used :
cp : stores the cost price
sp : stores the selling price
pr : stores the profit made
percentProfit : stores the calculated profit in percentage

NODIA

84. Write a program using Java, to print the real quotient obtained by dividing two integer numbers. Also, print the remainder obtained after dividing them.

Ans : SQP 2023

```
class Real {
    public static void main() {
        int num1, num2, rem;
        double q;
        num1 = 5;
        num2 = 50;
        q = (double)num1 / num2;
        rem = num1 % num2;
      System.out.println("The result
is " + q);
       System.out.println("Remainder
= " + rem);
    }
}
```

Variables used :
num1, num2 : the numbers to divide
q : stores the quotient
rem : stores the remainder

85. Write a program using Java to print the real quotient obtained by dividing two integer numbers. Also, print the remainder obtained after dividing them.

Ans : MAIN 2021

```
import java.util.Scanner;

class QuotientRemainder {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Input two integers
       System.out.println("Enter the
first integer (dividend) : ");
        int dividend = sc.nextInt();
       System.out.println("Enter the
second integer (divisor) : ");
        int divisor = sc.nextInt();

          // Calculate quotient and
remainder
         int quotient = dividend /
divisor;
         int remainder = dividend %
divisor;

        // Print the results
        System.out.println("Quotient
: " + quotient);
       System.out.println("Remainder
: " + remainder);

        sc.close();
    }
}
```

86. Write a program using Java to calculate and display the sum and difference of two integer numbers entered by the user.

Ans : SQP 2022

```
import java.util.Scanner;

class Sum Difference {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Input two integers
       System.out.println("Enter the
first number : ");
        int num1 = sc.nextInt();
       System.out.println("Enter the
second number : ");
        int num2 = sc.nextInt();

      // Calculate sum and difference
        int sum = num1 + num2;
        int difference = num1 - num2;

        // Print the results
         System.out.println("Sum : "
+ sum);
       System.out.println("Difference
: " + difference);

        sc.close();
    }
}
```

87. Write a Java class Time with member variables hours, minutes, and seconds. Implement a main method in the class that initializes two time instances (t1 and t2) with specific values and calculates their total time. Display the result in the format :

NODIA

totHours : totMinutes : totSeconds.

Ans : COMP 2024

```
class Time {
    int hours, minutes, seconds;

    public static void main() {
        Time T1 = new Time();
        Time T2 = new Time();
          int totHours, totMinutes,
totSeconds;
        int h, m, s;

        T1.hours = 10;
        T1.minutes = 40;
        T1.seconds = 35;

        T2.hours = 6;
        T2.minutes = 45;
        T2.seconds = 29;

        s = T1.seconds + T2.seconds;
        int s1 = s % 60;
        int s2 = s / 60;

         m = T1.minutes + T2.minutes
+ s2;
        int m1 = m % 60;
        int m2 = m / 60;

       h = T1.hours + T2.hours + m2;

        totHours = h;
        totMinutes = m1;
        totSeconds = s1;

         System.out.println(totHours
+ "   : " + totMinutes + "   : " +
totSeconds);
    }
}
```

```
int m1 = m % 60;
int m2 = m / 60;
h = T1.hours + T2.hours + m2;
totSeconds = s1;
totMinutes = m1;
totHours = h;

System.out.println("Total       time
is:");
System.out.println(totHours + ":" +
totMinutes + ":" + totSeconds);
}
}
```

88. Write a Java class Rectangle with member variables length and breadth. Implement a main method in the class that initializes two rectangle instances (r1 and r2) with specific values and calculates their total area and perimeter. Display the results in the format:
Total Area : <value>, Total Perimeter : <value>

Ans : SQP 2005

```
class Rectangle {
    int length, breadth;

    public static void main(String[]
args) {
      Rectangle r1 = new Rectangle();
      Rectangle r2 = new Rectangle();

        r1.length = 10;
        r1.breadth = 5;

        r2.length = 15;
        r2.breadth = 8;

           int area1 = r1.length *
r1.breadth;
           int area2 = r2.length *
r2.breadth;

       int totalArea = area1 + area2;

          int perimeter1 = 2 * (r1.
length + r1.breadth);
          int perimeter2 = 2 * (r2.
length + r2.breadth);

      int totalPerimeter = perimeter1
+ perimeter2;
           System.out.println("Total
Area : " + totalArea);
           System.out.println("Total
Perimeter : " + totalPerimeter);
    }
}
```

89. Write a main method for the class Time to calculate and display the difference between two given times in hours, minutes, and seconds.

Ans : COMP 2015

```
class Time2 {
    int hours, minutes, seconds;

    public static void main() {
         Time2 T1 = new Time2(); /*
Creating object */
         Time2 T2 = new Time2(); /*
```

Downloaded from www.pdf.tube

```
Creating object */

        int totSeconds1, totSeconds2;

       T1.hours = 05; /* Initializing
values of first time */
        T1.minutes = 35;
        T1.seconds = 20;

       T2.hours = 12; /* Initializing
values of second time */

        T2.minutes = 45;
        T2.seconds = 35;

        totSeconds1 = T1.hours * 3600
+ T1.minutes * 60 + T1.seconds;

        totSeconds2 = T2.hours * 3600
+ T2.minutes * 60 + T2.seconds;

            int diff = totSeconds2 -
totSeconds1;
        int hr = diff / 3600;
        int m1 = diff % 3600;

        int min = m1 / 60;
        int sec = m1 % 60;

       System.out.println("Difference
in time is:");

          System.out.println(hr + "
hours:" + min + " minutes:" + sec +
" seconds");
    }
}
```

90. Define a main method for the class Rectangle that calculates and displays the area and perimeter of two rectangles based on their length and breadth.

Ans : MAIN 2011

```
class Rectangle {
    int length, breadth;

    public static void main(String[
] args) {
      Rectangle R1 = new Rectangle();
      Rectangle R2 = new Rectangle();
        R1.length = 10;
        R1.breadth = 5;
        R2.length = 15;
        R2.breadth = 8;
           int area1 = R1.length *
R1.breadth;
          int perimeter1 = 2 * (R1.
length + R1.breadth);
            int area2 = R2.length *
R2.breadth;
          int perimeter2 = 2 * (R2.
length + R2.breadth);
       System.out.println("Rectangle
1:");
            System.out.println("Area
: " + area1 + ", Perimeter : " +
perimeter1);

       System.out.println("Rectangle
2:");
            System.out.println("Area
: " + area2 + ", Perimeter : " +
perimeter2);
    }
}
```

NODIA

91. Define a main method for the class Circle that calculates and displays the area and circumference of two circles based on their radii. Use the formula for area Area $= \pi \times r^2$ and for circumference Circumference $= 2 \times \pi \times r$ Assume $\pi = 3.14$.

Ans : COMP 2024

```
class Circle {
    double radius;

    public static void main(String[]
args) {
        Circle C1 = new Circle();
        Circle C2 = new Circle();

        // Assigning radius for both
circles
        C1.radius = 7.0;
        C2.radius = 10.0;

            // Calculating area and
circumference for Circle 1
      double area1 = 3.14 * C1.radius
* C1.radius;
         double circumference1 = 2 *
3.14 * C1.radius;

            // Calculating area and
circumference for Circle 2
      double area2 = 3.14 * C2.radius
* C2.radius;
         double circumference2 = 2 *
3.14 * C2.radius;

          // Displaying results for
```

```
Circle 1
          System.out.println("Circle
1:");
          System.out.println("Area :
" + area1 + ", Circumference : " +
circumference1);

          // Displaying results for
Circle 2
          System.out.println("Circle
2:");
          System.out.println("Area :
" + area2 + ", Circumference : " +
circumference2);
    }
}
```

92. Define a main method for the class Triangle that calculates and displays the area and perimeter of two triangles based on their base, height, and sides. Use the formula for area Area = 1/2×base×height and for Perimeter = side1+side2+side3.

Ans : SQP 2000

```
class Triangle {
   double base, height, side1, side2,
side3;

    public static void main(String[]
args) {
        Triangle T1 = new Triangle();
        Triangle T2 = new Triangle();

            // Assigning values for
Triangle 1
        T1.base = 10.0;
        T1.height = 8.0;
        T1.side1 = 10.0;
        T1.side2 = 8.0;
        T1.side3 = 6.0;

            // Assigning values for
Triangle 2
        T2.base = 15.0;
        T2.height = 12.0;
        T2.side1 = 15.0;
        T2.side2 = 12.0;
        T2.side3 = 9.0;

            // Calculating area and
perimeter for Triangle 1
        double area1 = 0.5 * T1.base
* T1.height;
        double perimeter1 = T1.side1
+ T1.side2 + T1.side3;

            // Calculating area and
perimeter for Triangle 2
        double area2 = 0.5 * T2.base
* T2.height;
        double perimeter2 = T2.side1
+ T2.side2 + T2.side3;

          // Displaying results for
Triangle 1
        System.out.println("Triangle
1:");
            System.out.println("Area
: " + area1 + ", Perimeter : " +
perimeter1);

          // Displaying results for
Triangle 2
        System.out.println("Triangle
2:");
            System.out.println("Area
: " + area2 + ", Perimeter : " +
perimeter2);
    }
}
```

93. Define a main method for the class Square that calculates and displays the area and perimeter of two squares based on their side lengths. Use the formula for area Area = side^2 and for perimeter Perimeter = 4 × side

Ans : COMP 2015

```
class Square {
    double side;

    public static void main(String[]
args) {
        Square S1 = new Square();
        Square S2 = new Square();

       // Assigning side lengths for
both squares
        S1.side = 5.0;
        S2.side = 7.0;

            // Calculating area and
perimeter for Square 1
          double area1 = S1.side *
S1.side;
           double perimeter1 = 4 *
S1.side;
```

Click Here to Buy book on Amazon

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

```
            // Calculating area and
perimeter for Square 2
            double area2 = S2.side *
S2.side;
            double perimeter2 = 4 *
S2.side;

            // Displaying results for
Square 1
            System.out.println("Square
1:");
            System.out.println("Area
: " + area1 + ", Perimeter : " +
perimeter1);

            // Displaying results for
Square 2
            System.out.println("Square
2:");
            System.out.println("Area
: " + area2 + ", Perimeter : " +
perimeter2);
    }
}
```

94. Write a Java class Distance with member variables feet and inches. Implement a main method to initialize two instances (d1 and d2) with specific values and calculate their total distance. Display the result in the format : totFeet' totInches".

Ans : SQP 2019

```
class Distance {
    int feet, inches;

    public static void main(String[]
args) {
        Distance d1 = new Distance();
        Distance d2 = new Distance();

        d1.feet = 5;
        d1.inches = 9;

        d2.feet = 6;
        d2.inches = 11;

        // Calculating total inches
        int totalInches = d1.inches
+ d2.inches;
        int extraFeet = totalInches
/ 12;
            int remainingInches =
totalInches % 12;

        // Calculating total feet
          int totalFeet = d1.feet +
d2.feet + extraFeet;

        // Displaying the result
           System.out.println("Total
Distance : " + totalFeet + "' " +
remainingInches + "\"");
    }
}
```

95. Write a Java class Duration with member variables hours, minutes, and seconds. Implement a main method to initialize two durations (d1 and d2) and calculate their total time. Display the result in the format : hh:mm:ss.

Ans : SQP 2015

```
class Duration {
    int hours, minutes, seconds;

    public static void main(String[]
args) {
        Duration d1 = new Duration();
        Duration d2 = new Duration();

        d1.hours = 2;
        d1.minutes = 50;
        d1.seconds = 40;

        d2.hours = 1;
        d2.minutes = 45;
        d2.seconds = 35;

        // Calculate total seconds
        int totalSeconds = d1.seconds
+ d2.seconds;
      int carryMinutes = totalSeconds
/ 60;
            int remainingSeconds =
totalSeconds % 60;

        // Calculate total minutes
        int totalMinutes = d1.minutes
+ d2.minutes + carryMinutes;
        int carryHours = totalMinutes
/ 60;
            int remainingMinutes =
totalMinutes % 60;

        // Calculate total hours
         int totalHours = d1.hours +
d2.hours + carryHours;
```

NODIA

```
        // Display result
           System.out.println("Total
Duration : " + totalHours +
":" + remainingMinutes + ":" +
remainingSeconds);
    }
}
```

96. Write a Java class Weight with member variables kilograms and grams. Implement a main method to initialize two weights (w1 and w2) and calculate their total weight. Display the result in the format : totKg kg totG g.

Ans : COMP 2011

```
class Weight {
    int kilograms, grams;

    public static void main(String[]
args) {
        Weight w1 = new Weight();
        Weight w2 = new Weight();

        w1.kilograms = 4;
        w1.grams = 750;

        w2.kilograms = 3;
        w2.grams = 500;

        // Calculate total grams
         int totalGrams = w1.grams +
w2.grams;
      int extraKilograms = totalGrams
/ 1000;
      int remainingGrams = totalGrams
% 1000;

        // Calculate total kilograms
    int totalKilograms = w1.kilograms
+ w2.kilograms + extraKilograms;

        // Display result
           System.out.println("Total
Weight : " + total Kilograms + " kg "
+ remaining-grams + " g");
    }
}
```

97. Write a program using Java to calculate and display the remainder and absolute difference of two integer numbers entered by the user.

Ans : MAIN 2020

```
import java.util.Scanner;

class RemainderAbsoluteDifference {
    public static void main(String[
] args) {
     Scanner sc = new Scanner(System.
in);

        // Input two integers
       System.out.println("Enter the
first integer : ");
        int num1 = sc.nextInt();

       System.out.println("Enter the
second integer : ");
        int num2 = sc.nextInt();

         // Calculate remainder and
absolute difference
        int remainder = num1 % num2;
           int absoluteDifference =
Math.abs(num1 - num2); // Absolute
difference

        // Display results
       System.out.println("Remainder
: " + remainder);
        System.out.println("Absolute
Difference : " + absoluteDifference);

        sc.close();
    }
}
```

98. Write a program that accepts the name, address, amount of purchase, and type of purchase (L for Laptop and D for Desktop) by a customer. Based on the purchase amount and type, compute and print the net amount to be paid. Use the following discount rates:

Purchase Amount (₹)	Discount on Laptop	Discount on Desktop PC
0 – 25000	0.0%	5.0%
25001 – 57000	5.0%	7.5%
57001 – 100000	7.5%	10.0%
More than 100000	10.0%	15.0%

Ans : SQP 2010

```
import java.util.Scanner;

class SeasonalDiscount {
    public static void main(String[]
```

Downloaded from www.pdf.tube

```
args) {
      Scanner sc = new Scanner(System.
in);

        // Input customer details
             System.out.print("Enter
Customer Name : ");
        String name = sc.nextLine();
             System.out.print("Enter
Address : ");
      String address = sc.nextLine();
       System.out.print("Enter Amount
of Purchase : ");
     double amount = sc.nextDouble();
        System.out.print("Enter Type
of Purchase (L for Laptop, D for
Desktop) : ");
            char type = sc.next().
charAt(0);

        double discount = 0;

         // Determine discount based
on amount and type
         if (type == 'L' || type ==
'l') { // Laptop
               if (amount <= 25000)
discount = 0;
           else if (amount <= 57000)
discount = 5;
          else if (amount <= 100000)
discount = 7.5;
            else discount = 10;
       } else if (type == 'D' || type
== 'd') { // Desktop
               if (amount <= 25000)
discount = 5;
           else if (amount <= 57000)
discount = 7.5;
          else if (amount <= 100000)
discount = 10;
            else discount = 15;
        }

       // Calculate discount and net
amount
            double discountAmount =
(discount / 100) * amount;
         double netAmount = amount -
discountAmount;

        // Output results
      System.out.println("\nCustomer
Details:");
        System.out.println("Name : "
+ name);
         System.out.println("Address
: " + address);
       System.out.println("Amount of
Purchase : ₹" + amount);
        System.out.println("Discount
Applied : " + discount + "%");
      System.out.println("Net Amount
to Pay : ₹" + netAmount);

        sc.close();
    }
}
```

99. Write a program to accept a character from the user and print whether it is a vowel, digit, space, consonant, or special character.

Ans : SQP 2008

```
import java.util.Scanner;

class CharacterCheck {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Input a character
          System.out.print("Enter a
character : ");
       char ch = sc.next().charAt(0);

        // Check the character type
       if (ch == 'a' || ch == 'e' ||
ch == 'i' || ch == 'o' || ch == 'u' ||
            ch == 'A' || ch == 'E' ||
ch == 'I' || ch == 'O' || ch == 'U') {
            System.out.println("It is
a vowel.");
             } else if (Character.
isDigit(ch)) {
            System.out.println("It is
```

NODIA

```
a digit.");
            } else if (Character.
isWhitespace(ch)) {
          System.out.println("It is
a space.");
            } else if (Character.
isLetter(ch)) {
          System.out.println("It is
a consonant.");
        } else {
          System.out.println("It is
a special character.");
        }

        sc.close();
    }
}
```

100. What is the use of a break statement followed by a label? Provide an example.

Ans : COMP 2005

The break statement followed by a label allows you to exit from an outer loop, even if the break is called within an inner loop.

```
public class BreakWithLabel {
    public static void main(String[]
args) {
        outerLoop:
       for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= 5;
j++) {
                if (j == 3) {
                    break outerLoop;
// Exits the outer loop
                }
               System.out.println("i
= " + i + ", j = " + j);
            }
        }
          System.out.println("Exited
the loops.");
    }
}
Output :
i = 1, j = 1
i = 1, j = 2
Exited the loops.
```

101. State the output of the following Java programs.

(i)

```
for (int i = 0; i < 10; i = i + 2) {
    System.out.println(++i);
}
```

(ii)

```
for (int i = 12; i > 5; i = i - 2) {
    System.out.println();
    System.out.println(i);
}
```

(iii)

```
int i, sum;
for (i = 5, sum = 1; sum < 10; i++) {
    sum += i;
}
System.out.println("sum is = " +
sum);
```

(iv)

```
int a = 0, i = 4;
while (i < 10) {
    a = ++i + i * 2;
}
System.out.println("a = " + a + " i
= " + i);
```

Ans : MAIN 2001

(i) 1
4
7

(ii) 12
10
8
6

(iii) sum is = 15

(iv) a = 30 i = 10]

102. A Pronic number is a number that is the product of two consecutive integers, i.e., $n=x\times(x+1)$, where x is an integer.

Ans : COMP 2002

```
import java.util.Scanner;

class PronicNumber {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Input a number
          System.out.print("Enter a
number : ");
```

Click Here to Buy book on Amazon

```
        int num = sc.nextInt();
        boolean isPronic = false;

         // Check if the number is a
Pronic number
         for (int i = 0; i * (i + 1)
<= num; i++) {
            if (i * (i + 1) == num) {
                 isPronic = true;
               System.out.println(num
+ " is a Pronic number.");
               System.out.println("It
is the product of " + i + " and " +
(i + 1) + ".");
                 break;
             }
        }

        if (!isPronic) {
            System.out.println(num +
" is not a Pronic number.");
        }

        sc.close();
    }
}
```

www.nodia.in

Downloaded from www.pdf.tube

CHAPTER 3

USER DEFINED METHODS

SUMMARY

1. METHOD OR FUNCTION

A method, also known as a function, is a collection of programming statements grouped under a specific name to perform a particular task. It is often referred to as a sub-program, representing a smaller, independent part of a larger program.

1.1 Use or Advantages of Functions

Functions provide a structured way to organize code, making programs more modular and easier to manage. They enhance code reusability, readability, and maintainability.

(i) Reuse of code : The code written once as a function can be called whenever required. Thus, repetitive writing of the code is avoided and once defined, the function is reused.

(ii) Easy debugging : A complex program is divided into simpler tasks and written using functions. It becomes easier to find and rectify an error in a small module of the program rather than the entire complex program.

(iii) Abstraction : To call or use a method in our program, we need to know only the following things:

1. Name of method
2. What the method does
3. What should be the arguments
4. What result does the method return

We do not need to know how the method works.

2. CLASSIFICATION OF FUNCTIONS

There are broadly two categories of functions:

(i) Library functions

(ii) User-define d functions

2.1 Library Functions

The functions that are predefined by the compiler are known as library functions. We have to only mention the functions. We have to only mention function prototype.

For example : Math.sqrt(), String.length(), and System.out.println() in Java. etc.

2.2 User-defined Functions

Programmers can define their own functions and call them as and when required. For example, a function to calculate the area of a circle or to check if a number is prime.

3. FUNCTION PROTOTYPE

A function prototype is the first statement of a function definition that specifies the functions return type, name, and parameters data types and variables. It acts as a blueprint for the function, providing essential details about its usage.

General format (or Syntax) :

```
AccessSpecifier                ReturnType
FunctionName(Parameter list);
```

Example -

public int calculateSum(int num1, int num2);

Description:

(i) AccessSpecifier : public determines the visibility of the function.

(ii) ReturnType : int specifies that the function returns an integer value.

(iii) FunctionName : calculateSum is the name of the function.

(iv) Parameter list : (int num1, int num2) specifies the input parameters data types and names.

4. FUNCTION BODY

The function body contains the Java statements that define the behavior or actions performed by the function. It is written within curly braces {} after the function prototype and includes the logic to achieve the functions purpose.

Example :

```
void show(int a)
{
```

NODIA

```
    System.out.println("Number is : " + a);
}
```

(i) void : Indicates that the function does not return a value.

(ii) show(int a) : The function prototype specifies the name and parameter(s).

(iii) Function Body : Contains the statements that execute when the function is called.

Note

If a function is designed to return a value, the last statement in the function body must be a return statement.

```
int add(int x, int y)
{
    return x + y;
}
```

5. FUNCTION SIGNATURE

The combination of a functions name and its parameter list is referred to as the functions prototype and is commonly known as its signature.

6. WAYS TO DEFINE A METHOD

Function definition can be written in various ways as given below

(i) Method with no returning value and no parameter.Example

```
void display() {
 System.out.println("Hello, World!");
}
```

(ii) Method with a returning value and no parameter. Example

```
int getNumber() {
    return 42;
 }
```

(iii) Method with no returning value and with parameters. Example

```
void printMessage(String message) {
   System.out.println(message);
}
```

(iv) Method with a returning value and with parameters. Example

```
int add(int a, int b) {
        return a + b;
        }
```

7. WAYS TO CALL OR INVOKE THE METHODS

Methods in Java can be invoked in two primary ways :

(i) Call by Value : In this approach, a copy of the actual parameters value is passed to the method. Changes made to the parameter inside the method do not affect the original value.

(ii) Call by Reference : In this approach, the reference (memory address) of the actual parameter is passed to the method. Changes made to the parameter inside the method directly affect the original value.

8. ACTUAL PARAMETERS

Actual parameters are the variables or values specified in the function call statement. For instance, in a function call like add(x, y), the variables x and y are the actual parameters. During the call by value process, the values of the actual parameters are copied to the corresponding formal parameters in the function definition. Separate memory locations are allocated for the actual and formal variables, ensuring that changes made to the formal parameters do not affect the actual parameters.

9. FORMAL PARAMETERS

The arguments (variables) in the first line of definition of the function are called formal parameters

10. RETURNING A VALUE FROM THE FUNCTION

Formal parameters are the variables specified in the functions definition to receive values from the actual parameters during a function call. These arguments are declared in the first line of the function definition and act as placeholders for the data passed to the function. For example, in the function definition void add(int a, int b), a and b are the formal parameters that will hold the values of the actual parameters during the function call

11. RETURN STATEMENT

General formats

(i) Return variable;

(ii) Return expression;

(iii) Return constant value;

Example :

```
return a;
return a*a+10;
return 50;
return 0;
return true;
```

Click Here to Buy book on Amazon

12. REFERENCE VARIABLE OR REFERENCE PARAMETER

A reference variable is another name of a memory location. Actually, a memory location which stores the actual value can be accessed by different references (reference variables). Any change of value in reference variable will be reflected in the original memory location

In Java, all reference data types objects and arrays are passed to a function by reference

13. CALLING THE FUNCTION BY VALUE

The statements with the following general format are used to call a function. The statement to be used depends upon the way the function is defined. In this process, different memory locations are created for actual and formal parameters. The value of actual parameters are copied to formal parameters

(i) If the function does not return a value :

General format

```
function_name(actual parameters);
```

Example`sum(a, b);`

(ii) If the function returns a value

General format

```
return_type variablename = function_
name(actual parameters);
```

Example`int a = sum(x, y);`

14. CALLING A FUNCTION BY REFERENCE

In call by reference, the formal parameters act as reference variables. When an object is passed to a method, only its reference (memory address) is passed, not a separate copy of the object. Any operation performed on the reference parameter directly affects the original object because the reference points to the same memory location. As a result, changes made to the formal parameters are reflected in the actual parameters. To make the operation effective on the object, the method must manipulate the reference parameter object through its member variables. This approach is commonly used in Java for passing objects and arrays to methods.

Here is an example program demonstrating call by reference to exchange the values of two objects:

```
public class Values
{
    int x;
}

class Swap
{
     static void exchange(Values a,
Values b)
    {
        Values temp = new Values();
        temp.x = a.x;
        a.x = b.x;
        b.x = temp.x;
    }

     public static void main(String
a[])
    {
        Values ob1 = new Values();
        Values ob2 = new Values();
        ob1.x = 10;
        ob2.x = 20;

        System.out.println("Original
ob1 : " + ob1.x);
        System.out.println("Original
ob2 : " + ob2.x);
        exchange(ob1, ob2);
         System.out.println("Changed
ob1 : " + ob1.x);
         System.out.println("Changed
ob2 : " + ob2.x);
    }
}
```

15. METHOD OVERLOADING

When more than one method has the same name but different list of parameters (number of parameters or data type of parameters), they are said to be overloaded. It means that the parameter list of each method should be unique. It is a practical implementation of the concept of polymorphism.

Example of function overloading :

```
float sum(float a, float b) {
    return a + b;
}

int sum(int a, int b, int c) {
    return a + b + c;
}
```

16. CREATING OBJECTS

In Java, objects are instances of a class and are created using the new keyword. The object serves as a reference to access the data (attributes) and behaviors (methods) defined in the class. The process involves calling the class constructor, which initializes the object and allocates memory

NODIA

for it at runtime. Objects are fundamental to Object-Oriented Programming (OOP), allowing classes to be instantiated and used in a program.

```
ClassName Object_variable = new ClassName();
```

17. MEMBER METHODS

Member methods are non-static methods that are associated with the objects of a class. These methods operate on the instance variables (member variables) defined in the class and can only be called using an object of the class. Member methods typically define the behavior or functionality that objects of the class can perform. They rely on the specific data of the object that invokes them.

Example of Member Methods

```
class Test
{
    int a, b;
    void sum()
    {
        System.out.print(a + b);
    }
}
```

18. NON-MEMBER METHODS

Non-member methods are methods declared with the static keyword. These methods are shared across all objects of a class and are not tied to any specific object. Since they are independent of objects, they can be called directly using the class name, without requiring an instance. Non-member methods typically work with static variables or perform operations that do not depend on instance-specific data.

Example of Non-Member Methods

```
static void checkSum()
{
    int a = 10, b = 20;
    System.out.print(a + b);
}
```

19. CALLING OF A MEMBER METHOD IN A NON-MEMBER METHOD

```
public static void main()
{
    // Create object // object name be obj
    // Associate member method with object
    obj.sum();
}
```

20. CALLING OF A MEMBER METHOD IN ANOTHER MEMBER METHOD

```
void example()
{
    // Associating with object not needed
    sum();
}
```

21. CALLING OF A NON-MEMBER METHOD

```
public static void main()
{
    checkSum();  // No association with object
}
```

22. PURE FUNCTION

A function that only returns a value without bringing about any change in the values being passed to it is called a pure function. In other words, there is no change in the state of the function

23. EXAMPLE OF A PURE FUNCTION

```
static boolean max(int a, int b)
{
    if (a == b) return true;
    else return false;
}
```

24. IMPURE FUNCTION

A function that brings about a change in the argument that it receives. Its arguments will always be reference types. It may or may not return a value. In other words, an impure function brings about a change in the state of the function. This change in state is called the side effect of calling an impure function

25. EXAMPLE OF AN IMPURE FUNCTION

```
static void count(Number num)
{
    num.counter = num.counter + 1;
}
/* The above function takes the argument num as an object of class Number. The counter variable of object num is incremented by 1 every time this function is called. */
```

Download INDIA's Best Study app NODIA

Downloaded from www.pdf.tube

MULTIPLE CHOICE QUESTION

1. What is an advantage of using functions in programming?

(a) Code repetition

(b) Abstraction

(c) Increased complexity

(d) Slower debugging

Ans : MAIN 2024

(b) Abstraction
Functions help abstract the complexity of code by encapsulating functionality into reusable blocks. This makes code more modular and maintainable.

2. What does the term "function signature" refer to?

(a) The body of the function

(b) The return type of the function

(c) The combination of the functions name and parameter list

(d) The access specifier of the function

Ans : SQP 2023

(c) The combination of the functions name and parameter list
A function signature uniquely identifies a function, typically including the name and parameter types, which are essential for distinguishing overloaded methods.

3. What is the correct syntax for creating an object in Java?

(a) ClassName objectName = ClassName();

(b) ObjectName = new ClassName();

(c) ClassName objectName = new ClassName();

(d) New ClassName objectName;

Ans : COMP 2022

(c) ClassName objectName = new ClassName();
This is the proper syntax for creating and initializing an object in Java using the new keyword.

4. Which of the following is an example of a non-member method?

(a) Void display()

(b) Static void checkSum()

(c) Void sum()

(d) Public void calculate()

Ans : MAIN 2021

(b) Static void checkSum()
Static methods belong to the class rather than an instance of the class, making them non-member methods in the context of objects.

5. What type of function brings about a change in its arguments and is referred to as an impure function?

(a) Pure function

(b) Static function

(c) Impure function

(d) Reference function

Ans : SQP 2000

(c) Impure function
An impure function modifies its arguments or depends on external state, producing side effects.

6. In Java, how are objects passed to methods?

(a) Call by Value

(b) Call by Reference

(c) Call by Name

(d) Call by Assignment

Ans : COMP 2002

(b) Call by Reference
In Java, object references are passed by value, but since the reference points to the same object, changes to the object affect the original.

7. Which of the following is a valid function prototype?

(a) Public void calculateSum()

(b) Public calculateSum(int a, int b)

(c) CalculateSum(int a, int b)

(d) Public int calculateSum(int a, int b)

Ans : MAIN 2003

(d) Public int calculateSum(int a, int b)
A function prototype must include the return type, method name, and parameter list.

8. Which of these statements is true for method overloading?

(a) Methods with the same name and identical parameter lists can be overloaded.

(b) Methods with the same name and different parameter lists are considered overloaded

(c) Overloaded methods must return the same data type

(d) Overloaded methods cannot have more than two parameters

NODIA

Ans : SQP 2004

(b) Methods with the same name and different parameter lists are considered overloaded
Method overloading occurs when methods have the same name but different parameter types or counts.

9. Which method call in the following options refers to calling a member method?

(a) ObjectName.sum()
(b) CheckSum()
(c) Main()
(d) New ClassName();

Ans : COMP 2005

(a) ObjectName.sum()
objectName.sum() because it refers to calling a member method of a class using an object. In Java, member methods are instance-specific methods, and they must be accessed through an object of the class. In this case, sum() is a member method of the class, and it is called using the syntax objectName.sum(), where objectName is an instance of the class.

10. What is the general format for calling a function that does not return a value?

(a) Return_type variableName = function_name(actual_parameters);
(b) Function_name(actual_parameters);
(c) Function_name;
(d) Function_name[parameters];

Ans : MAIN 2006

(b) Function_name(actual_parameters);
This is the correct way to call a function that does not return a value. The function is called using its name, followed by parentheses enclosing any required arguments.

11. Find the odd one out

(a) Method overloading
(b) Polymorphism
(c) Abstraction
(d) Multiple forms of same

Ans : MAIN 2009

(c) Abstraction
While overloading, polymorphism, and "multiple forms of the same" are related to polymorphism, abstraction deals with hiding implementation details.

12. Built-in functions are also called:

(a) Library methods
(b) User-defined methods
(c) Fixed methods
(d) All of the above

Ans : SQP 2007

(a) Library methods
Built-in functions are predefined methods provided by programming languages or their standard libraries. These functions are part of the language's library, and they are designed to perform common tasks such as mathematical operations, string manipulations, input/output handling, etc. Since they are part of the library, they are also referred to as library methods.

13. How many return statements can a method have and how many of them should get executed for the program to run successfully?

(a) 1, 1
(b) Many, 1
(c) 1, Many
(d) Many, Many

Ans : MAIN 2015

(b) Many, 1
A method can have multiple return statements, but only one is executed during a single call.

14. A method in which the method parameters can get modified is called a _____

(a) Pure method
(b) Impure method
(c) Virtual method
(d) None of the above

Ans : SQP 2010

(a) Pure method
An impure method is a method where the parameters passed to it can be modified. This means that the method may have side effects, such as changing the state of the arguments (if they are objects or passed by reference) or modifying global variables.

15. A method prototype has which of the following components?

(a) Name of method
(b) Return type of method
(c) List of formal parameters
(d) All of the above

Click Here to Buy book on Amazon

Ans : COMP 2011

(d) All of the above
A method prototype (also called a method declaration or signature in some contexts) defines the structure of a method without providing its implementation.

16. A method with no result or value to return has a return type as
(a) Null
(b) Void
(c) No return type
(d) Default

Ans : COMP 2008

(b) Void
In Java (and many other programming languages), when a method does not return any value, its return type is explicitly specified as void. The void keyword indicates that the method performs some actions but does not return a result to the caller.

17. When a reference is passed to a method as a parameter, which of the following statement is true?
(a) The reference parameter creates a copy of the variable passed to it
(b) Reference parameter refers to the same variable passed to it
(c) The reference parameter value can get changed but passing variable value does not get changed
(d) A reference can be of simple data type

Ans : MAIN 2012

(b) Reference parameter refers to the same variable passed to it
When a reference is passed to a method as a parameter, the method receives a reference to the actual object, not a copy of it. This means that any changes made to the object using the reference in the method will directly affect the original object outside the method. This behavior occurs because the reference parameter points to the same memory location as the original variable.

18. An operator used to invoke methods or variables using class name
(a) New
(b) Dot
(c) Relational
(d) Arithmetic

Ans : SQP 2013

(b) Dot
The dot operator (.) is used in Java to access methods or variables of a class. When used with a class name, it allows accessing static methods or static variables of the class. Similarly, when used with an object, it enables accessing instance methods or instance variables.

19. What would be the prototype of the below method for implementing method overloading?
int check(double x)
(a) Double check(int x, double y)
(b) Double check(double x)
(c) Both (a) and (b)
(d) Neither (a) nor (b)

Ans : SQP 2016

(a) Double check(int x, double y)
This has a different parameter list (two parameters instead of one), which makes it a valid overload. The return type (double) is irrelevant for overloading, but the parameters differ, so it satisfies the conditions for overloading.

20. The parameters that are used in the prototype of the method are called
(a) Actual parameters
(b) Formal parameters
(c) Virtual parameters
(d) Real parameters

Ans : COMP 2017

(b) Formal parameters
Formal parameters are the parameters listed in the method prototype or declaration. These are placeholders for the values that will be passed to the method when it is called. They are defined within the parentheses of the method signature and exist only within the scope of the method during its execution.

21. Which of the following java statement will compile successfully for defining a method to receive a list of 10 values?
(a) Void test-method (int a[10])
(b) Void test-method (int a[ ])
(c) Void test-method (int a)0
(d) Both (b) and (c)

Ans : COMP 2014

(d) Both (b) and (c)

NODIA

In Java, when defining a method that takes an array as a parameter, the array size is not explicitly specified in the method declaration. Java does not allow specifying the size of an array in the parameter list because the size of the array is determined at runtime, not at compile time.

22. The number of values that a method can return is

(a) 1

(b) 2

(c) 3

(d) 4

Ans : MAIN 2018

(a) 1

In Java, a method can directly return only one value. This is because the method's return type specifies the type of the single value it can return. When a return statement is executed, the method terminates and provides that single value back to the caller.

23. A student writes the following code to multiply a variable z by 5. He has written the following statement, which is incorrect:

```
z * 5;
```

What will be the correct statement?

A. z *= 5;

B. z = z * 5;

C. z * 5;

(a) Only A

(b) Only B

(c) Both A and B

(d) All the three

Ans : SQP 2019

(c) Both A and B

The given statement z * 5; is incorrect because it performs a multiplication operation but does not store the result anywhere. In Java, such a statement has no effect since the result of the multiplication is not assigned back to the variable or used elsewhere.

24. A student writes the following code to divide a variable a by 4. He has written the following statement, which is incorrect:

```
a = /4;
```

What will be the correct statement?

A. a /= 4;

B. a = a / 4;

C. a = /4;

(a) Only A

(b) Only B

(c) Both A and B

(d) All the three

Ans : COMP 2020

(c) Both A and B

The given statement a = /4; is incorrect because it is not valid syntax in Java. The / operator must be used between two operands, and the statement must assign the result to a variable correctly.

25. A student writes the following code to increment a variable n by 1. He has written the following statement, which is incorrect:

```
n + 1;
```

A. n++;

B. n += 1;

C. n = n + 1;

(a) Only A

(b) Only C

(c) All A, B, and C

(d) None of the above

Ans : MAIN 2021

(c) All A, B, and C

The statement written by the student, n + 1;, is incorrect because it simply evaluates the expression n + 1 but does not assign the result back to the variable n.

All three forms increment n by 1.

26. A student writes the following code to subtract 7 from a variable p.

Both p -= 7; and p = p - 7; are correct.

He has written the following statement, which is incorrect:

p = 7-;

What will be the correct statement?

A. p -= 7;

B. p = p - 7;

C. p = -7;

(a) Both A and B

(b) Only B

(c) Only C

(d) None of the above

Ans : SQP 2022

(a) Both A and B

All three statements double the value of q

NODIA

27. A student writes the following code to double the value of a variable q. He has written the following statement, which is incorrect:

```
q + q;
```

What will be the correct statement?

A. q *= 2;

B. q = q * 2;

C. q = q + q;

(a) Only A

(b) Only B

(c) Both A and B

(d) All A, B, and C

Ans : COMP 2023

(d) All A, B, and C

The student's statement q + q; is incorrect because it simply evaluates the sum of q + q but does not assign the result back to the variable q.

28. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to calculate the average of two numbers.

```
void average(int a, int b)
{
    int sum = 0;  // (1)
    int avg = 0;  // (2)
    avg = sum / 2;  // (3)
    sum = a + b;  // (4)
}
```

(a) (1) (3) (2) (4)

(b) (4) (1) (3) (2)

(c) (2) (1) (4) (3)

(d) (1) (4) (2) (3)

Ans : MAIN 2024

(d) (1) (4) (2) (3)

29. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to calculate the product of two numbers.

```
void multiply(int a, int b)
{
    int product = 0;  // (1)
    product = a * b;  // (2)
  System.out.println(product);// (3)
}
```

(a) (1) (2) (3)

(b) (2) (1) (3)

(c) (3) (1) (2)

(d) (1) (3) (2)

Ans : SQP 2000

(a) (1) (2) (3)

30. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to reverse two numbers using addition and subtraction.

```
void reverse(int x, int y)
{
 x = x + y;// (1)
 y = x - y;// (2)
 x = x - y;// (3)
}
```

(a) (1) (3) (2)

(b) (2) (1) (3)

(c) (3) (2) (1)

(d) (1) (2) (3)

Ans : COMP 2003

(d) (1) (2) (3)

31. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to initialize and display a variable.

```
java
void initialize()
{
 int num = 0;// (1)
 num = 10;// (2)
 System.out.println(num);// (3)
}
```

(a) (1) (3) (2)

(b) (2) (1) (3)

(c) (1) (2) (3)

(d) (3) (2) (1)

Ans : MAIN 2005

(c) (1) (2) (3)

32. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to swap the values of two variables without using a third variable.

```
{
a = a + b;// (1)
b = a - b;// (2)
a = a - b;// (3)
}
```

(a) (1) (2) (3)

(b) (3) (2) (1)

(c) (2) (1) (3)

(d) (1) (3) (2)

Downloaded from www.pdf.tube

Ans : SQP 2005

(a) (1) (2) (3)
a = a + b; combines a and b into a.
b = a - b; assigns the original value of a to b.
a = a - b; assigns the original value of b to a.

ASSERTION AND REASON

33. **Assertion (A) :** Non-member methods are tied to specific objects in Java.
Reason (R) : Non-member methods are declared using the static keyword and can be accessed without creating an object.

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : SQP 2024

(d) Assertion (A) is false, but Reason (R) is true.
In Java, non-member methods are not tied to specific objects, as they are declared using the static keyword and belong to the class rather than any instance. Static methods can be accessed without creating an object, which makes the reason true. However, the assertion that non-member methods are tied to specific objects is false. Therefore, Assertion (A) is false, but Reason (R) is true.

34. **Assertion (A) :** A method is also known as a function in Java.
Reason (R) : A method is a collection of programming statements grouped together under a specific name to perform a particular task.

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : COMP 2012

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
In Java, a method is also referred to as a function, as it performs tasks similar to functions in other programming languages. A method is a collection of programming statements grouped under a specific name to execute a particular task. It can take input parameters, return a value, and is central to achieving code re usability and modularity in Java. The assertion that a method is known as a function is true, and the reason provided accurately explains its purpose. Thus, both the assertion and the reason are correct, with the reason serving as the appropriate explanation for the assertion.

35. **Assertion (A) :** In Java, the return statement is used in functions that do not return any value.
Reason (R) : The return statement specifies the value to be returned by the function.

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : MAIN 2011

(d) Assertion (A) is false, but Reason (R) is true.
The assertion that "In Java, the return statement is used in functions that do not return any value" is false. In Java, the return statement is not required in methods with a void return type, as these methods do not return any value. However, a return statement can be used in void methods to terminate execution early, though it does not specify a value. The reason that "The return statement specifies the value to be returned by the function" is true. In methods with a non-void return type, the return statement is used to provide the value that the method returns to the caller. Therefore, the assertion is incorrect, but the reason is correct, making the correct evaluation Assertion (A) is false, but Reason (R) is true.

36. **Assertion (A) :** Functions help improve code modularity.
Reason (R) : Functions allow the same block of code to be reused multiple times, thus avoiding repetition.

NODIA

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : MAIN 2015

(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
Functions play a key role in improving code modularity by dividing a program into smaller, manageable, and independent units, each performing a specific task. This modularity enhances readability, maintainability, and debugging of code.

The reason provided, "Functions allow the same block of code to be reused multiple times, thus avoiding repetition," is also true and directly supports the assertion. By enabling code reuse, functions eliminate redundancy and contribute to modularity. Therefore, both the assertion and the reason are correct, and the reason provides a valid explanation for the assertion.

37. **Assertion (A) :** Library functions in Java are predefined by the compiler.
Reason (R) : Library functions like Math.sqrt() and String.length() do not require the user to define their functionality.
(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : SQP 2011

(d) Assertion (A) is false, but Reason (R) is true. Reason is false because library functions in Java are not defined by the compiler. Instead, they are part of the Java Standard Library (JSL) and provided by the Java API (Application Programming Interface), which is included in the Java Development Kit (JDK). The compiler is responsible for compiling code but does not define library functions. Reason is true because these functions are pre-implemented and readily available in Java libraries for direct use. Developers can call them without needing to write their implementations.

38. **Assertion (A) :** Function prototypes act as blueprints for functions in Java.
Reason (R) : A function prototype specifies the return type, name, and parameters data types in a function.
(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : COMP 2017

(d) Assertion (A) is false, but Reason (R) is true. In Java, function prototypes do not exist. Unlike C or C++, where prototypes declare a function's return type, name, and parameters before its definition, Java requires methods to be fully defined with their signature and implementation within a class.

The statement that function prototypes specify the return type, name, and parameters is true for languages like C or C++, but it does not apply to Java. Thus, the assertion that "Function prototypes act as blueprints for functions in Java" is false, while the reason provided is true. Therefore, Assertion (A) is false, but Reason (R) is true.

39. **Assertion (A) :** Method overloading is an example of polymorphism.
Reason (R) : Method overloading allows methods to have the same name but different parameter lists.
(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : COMP 2010

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).

Click Here to Buy book on Amazon

Method overloading is an example of compile-time polymorphism in Java. It allows methods in the same class to share the same name but differ in parameter lists, enabling varied behavior based on arguments.

The assertion that "Method overloading is an example of polymorphism" is true, and the reason that "Method overloading allows methods to have the same name but different parameter lists" correctly explains it. Thus, both the assertion and reason are true, with the reason being the correct explanation.

40. **Assertion (A) :** Call by value creates separate memory locations for actual and formal parameters.
Reason (R) : In call by value, the values of the actual parameters are copied into the formal parameters.

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : MAIN 2010

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
In call by value, the values of actual parameters are copied into formal parameters, resulting in separate memory locations for each. Changes to formal parameters do not affect actual parameters because they operate on different memory. Therefore, both the assertion and reason are true, and the reason explains the assertion correctly.

41. **Assertion (A) :** Objects in Java are created using the new keyword.
Reason (R) : The new keyword allocates memory for the object and calls its constructor.

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : SQP 2019

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
In Java, objects are created using the new keyword, which is responsible for allocating memory for the object and calling its constructor to initialize it. This process ensures that the object is properly instantiated with the required attributes and behaviors defined by its class. The assertion that "Objects in Java are created using the new keyword" is true, and the reason that "The new keyword allocates memory for the object and calls its constructor" accurately explains this process. Therefore, both the assertion and the reason are true, and the reason serves as the correct explanation for the assertion.

42. **Assertion (A) :** A pure function does not change the state of the arguments passed to it.
Reason (R) : A pure function only performs operations on the arguments and returns a value without side effects.

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
(b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
(c) Assertion (A) is true, but Reason (R) is false.
(d) Assertion (A) is false, but Reason (R) is true.

Ans : COMP 2018

(a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
A pure function is defined as a function that does not produce side effects, meaning it does not alter the state of its arguments or any external variables. Instead, it solely operates on the arguments passed to it and returns a value. The assertion that "A pure function does not change the state of the arguments passed to it" is true because pure functions ensure immutability and side-effect-free behavior. The reason that "A pure function only performs operations on the arguments and returns a value without side effects" correctly explains the concept of a pure function. Hence, both the assertion and reason are true, and the reason is the correct explanation.

NODIA
Download INDIA's Best Study app NODIA

TWO MARKS QUESTIONS

43. (i) Write two advantages of using functions in a program
(ii) Differentiate between static and non-static data members

Ans : MAIN 2024

(i) Advantages of using functions are
(a) Re usability of code
(b) Easy debugging and testing of programs
(ii) Static members are not associated with objects, whereas non-static members are associated with objects

44. Write a function prototype of the following
(i) A function PosChar which takes a string argument and a character argument and returns an integer value
(ii) Function that returns a double value and takes three double parameters x, y, and z as its parameters.

Ans : COMP 2021

(i) int PosChar(String st, char a);
(ii) double funcQ7(double x, double y, double z);

45. Write a function prototype of the following
(i) Function that returns a boolean value and takes an integer variable as its argument
(ii) A static method named calculator that accepts two short type variables as its arguments and returns an integer value.

Ans : MAIN 2022

(i) boolean funcQ8(int x);
(ii) static int calculator(short x, short y);

46. Write a function prototype of the following
(i) A function divide that takes two integer values and returns the quotient of double type.
(ii) An overloaded function that takes two double values as formal parameters and returns the quotient of double type

Ans : COMP 2024

(i) Double divide(int x, int y);
(ii) Double divide(double m, double n);

47. What are the two ways of invoking functions?

Ans : SQP 2000

Functions can be invoked in two ways:
(i) Call by Value : A copy of the actual arguments is passed to the function, so changes made inside the function do not affect the original values.
(ii) Call by Reference : The memory address of the arguments is passed, allowing the function to modify the original values directly

48. What are formal parameters?

Ans : MAIN 2001

Formal parameters are the variables declared in a function's definition to receive values or data passed to the function when it is called. They act as placeholders for the actual parameters (arguments) and are only accessible within the function. For example:

```
void add(int a, int b) {  // 'a' and
'b' are formal parameters
    printf("%d", a + b);
}
```

49. Write the general format of defining a function in a java class

Ans : SQP 2023

```
[access specifier][static] returntype
functionname(formal parameters)
{
    statements; // Function body
}
```

50. What are actual parameters?

Ans : COMP 2002

Actual parameters are the values or variables passed to a function when it is called. They are the real inputs provided to the function and are assigned to the corresponding formal parameters in the function definition.
For example:

```
add(5, 10);   // '5' and '10' are
actual parameters
```

51. Differentiate between actual parameter and formal parameter

Ans : SQP 2003

Actual parameter : The arguments passed to the function while calling the function
Formal parameter : The parameters that receive the values from the actual parameters
Example

```
void sum(int a, int b) {
```

Downloaded from www.pdf.tube

```
    int c = a + b;
    System.out.print(c);
}

void calling() {
    int x = 5, y = 10;
    sum(x, y);
}
```

Here, a and b are formal parameters and x and y are actual parameters

52. (i) What is meant by private visibility of a method?
(ii) Explain the function of a return statement

Ans : MAIN 2006

(i) A method is given a private access if its scope is to be limited to the class in which it is defined. The other classes will not be able to invoke the private method of a class

(ii) Returning a value from the function
If the prototype of the function shows a return type other than void, then its body must have a return statement as the last statement of the function. A function can return only ONE value
return statement
General formats
(a) return variable; For example : return a;
(b) return expression; For example : return a*a+10;
(c) return constant value; For example

```
return 50;
return 0;
return true;
```

53. (i) What is the role of the keyword void in declaring functions?
(ii) If a function contains several return statements, how many of them will be executed?
(iii) Which OOP principle implements function overloading?

Ans : COMP 2005

(i) Void indicates that the function does not return any value

(ii) Only one return statement will be executed

(iii) Polymorphism

54. What is method overloading?

Ans : SQP 2007

When more than one function has the same name but the type and number of arguments are different, they are said to be overloaded. Thus, the parameter list of each function should be unique. Function overloading implements the concept of polymorphism
Example :

```
float sum(float a, float b)
{
    return a + b;
}

int sum(int a, int b, int c)
{
    return a + b + c;
}
```

Here, in both methods, the method name is the same (sum), but the arguments and return type are different

55. How are the following passed?
(i) Primitive types
(ii) Reference types

Ans : MAIN 2004

(i) Primitive types are passed by value to the methods

(ii) Reference types are passed by reference, i.e., formal and actual arguments refer to the same memory locations

56. Read the following overloaded method prototypes and answer the questions that follow

```
void calculate(int x, double y);
void calculate(double x, double y);
```

(i) Write the Java statement to invoke the method calculate using actual parameters, a = 20, b = 45.52
(ii Write the Java prototype to overload the method calculate to return a double result and accept three double parameters

Ans : COMP 2008

(i) obj.calculate(20, 45.52);where obj is an object of the class in which calculate methods are defined

(ii)

```
double calculate(double x, double y, double z);
```

57. Read the following overloaded method prototypes and answer the questions that follow :

```
static Number calcMaths(Number num)
{
```

NODIA

```
    num.a++;
    num.b++;
    return num;
}

static                          ApplyMaths
calcMaths(ApplyMaths num)
{
    ApplyMaths ob = new ApplyMaths();
    ob.x = +num.x;
    return ob;
}
```

(i) What is the first method calcMaths computing?

(ii) What are the returning types of the overloaded method calcMaths?

(iii) Write the statement to call the second method definition of calcMaths

Ans : MAIN 2009

(i) The calcMaths method increments the values of data members a and b for the object num by 1

(ii) calcMaths returns objects, so their returning type is the class name (ApplyMaths) of which it is an object

(iii)

```
ApplyMaths obj = new ApplyMaths();
ApplyMaths obj2 = new ApplyMaths();
obj2 = calcMaths(obj);
/* Static function calcMaths can be
directly invoked in another static
method */
```

58. Why is main method defined as a static method in a Java class?

Ans : SQP 2010

main() is the first method that gets called in a Java class (program). The keyword static makes the Java interpreter call main method without the need of creating any object. All non-static methods can be called only by writing object name and method name separated by a dot

59. What is the difference between pure and impure methods?

Ans : MAIN 2011

Pure function : A function that only returns a value without bringing about any change in the values being passed to it is called a pure function. In other words, there is no change in the state of the function

Example of a pure function

```
static boolean max(int a, int b)
{
    if (a == b) return true;
    else return false;
}
```

Impure function : A function that brings about a change in the argument that it receives. Its arguments will always be reference types. It may or may not return a value. In other words, an impure function brings about a change in the state of the function. This change in state is called the side effect of calling an impure function

Example of an impure function

```
static void count(Number num)
{
    num.counter = num.counter + 1;
}
/* The above function takes the
argument num as an object of class
Number
The counter variable of object num
is incremented by 1 every time this
function is called. */
```

60. Write the outputs of the following segments of Java program

(i)

```
class Output1 {
    boolean var;
    Output1() {
        var = true;
    }
     public static void main(String
args[]) {
        Output1 out = new Output1();
        System.out.println(out.var);
    }
}
```

(ii)

```
class Assumption {
    private char s;
    private static int a;
    public int total;
    Assumption() {
        a = 0;
        s =  ;
        total = 0;
    }
```

Click Here to Buy book on Amazon

```
    public static void main() {
            Assumption assump = new
Assumption();
       System.out.println("Character
is " + assump.s);
        System.out.print("total is "
+ assump.total);
       System.out.print("a is " + a);
    }
}
/* End of main method */
/* End of class */
```

Ans : MAIN 2010

(i) Output : true

(ii) Output Character is total is 0 a is 0

61. Read the given segments of program and answer the questions that follow :

```
public boolean func(int a) {
 int c = a % a + ++a - ++a;
 System.out.println(c);
 boolean d = false;
 return d;
 }
```

(i) Write one statement that calls the above method in another non-static method named getValue()

(ii) What will be the output of the above-defined method for func(20)?

Ans : SQP 2013

(i) boolean var = func(10);

(ii) Output : -(i) This method returns false

62. Read the following impure method definitions and answer the questions that follow

```
void Compute(ApplyMaths a, ApplyMaths
b) {
 a.c = a.c + 1;
 b.d = a.c + 10;
}
```

(i) Write the reference data types in the above function

(ii) Name the class as you read in the above definition

(iii) Name the member variables of this class

(iv) What is the method computing?

Ans : MAIN 2014

(i) ApplyMaths a and ApplyMaths b

(ii) ApplyMaths

(iii) c and d

(iv) It increments the data member c of object a by 1 and the data member d of object b stores 10 added to the changed value of a.c

63. Read the following impure method definitions and answer the questions that follow :

```
static int solve(int a, Number
                                num) {
num.value *= a;
return num.value;
}
```

(i) State the output for the following function call
int c = solve(10, ob); (if all member variables for this object ob are initialized as 0)

(ii) Name the member variable as read in the method

(iii) What is the expected data type of this member variable?

(iv) Name the reference data type and the primitive data types as read from the function definition

Ans : COMP 2015

(i) num.value *= a;
Here, the value of a is passed as 10, and num object is passed with 0 value for all its data members (num.value)
So, num.value = num.value * a
= 0 * 10 = 0
Output : 0 for num.value

(ii) Member variable : value

(iii) Expected data type : int

(iv) Reference data type : Number
Primitive data types : int.

64. Read the given segments of program and answer the questions that follow :

```
public void displayMessage(void)
{
System.out.print("NET_SAVVY");
 System.out.print("--FOLLOW
            NETIQUETTES--");
 System.out.println();
 System.out.println("*");
}
```

(i) What happens if the following statement is written in public static void main() to call the function : displayMessage();

(ii) Write the output of the method

Ans : COMP 2012

Downloaded from www.pdf.tube

(i) Error : Non-static method displayMessage() cannot be invoked in a static method main() without associating it with an object

```
Correction
    className obj = new className();
    obj.displayMessage();
```

(ii) Output

```
        plaintext
NET_SAVVY -FOLLOW NETIQUETTES--
*****************
```

65. Define a Java method volSphere to compute the volume of a sphere given as $\frac{4}{3}\pi r^3$ for a sphere of radius 5 cm.

Ans : SQP 2016

```
public class Q3 {
   public static double volSphere(int
r) {
        return 4.0 / 3.0 * 3.14 * r
* r * r;
    }

    public static void main() {
        System.out.println("Vol = "
+ volSphere(5));
    }
}
```

66. Write a complete Java program to calculate the average of three numbers passed to a method AVG, which returns the result to main()

Ans : MAIN 2017

```
public class Q3 {
    public double AVG(int a, int b,
int c) {
        return (a + b + c) / 3.0;
    }

    public static void main() {
        Q3 obj = new Q3();
int num1 = 5, num2 = 10, num3 = 15;
System.out.println("The average of
5, 10 and 15 = " + obj.AVG(num1,
num2, num3));
}
}
```

Output

```
The average of 5, 10, and 15 = 10.0
4
```

Download INDIA's Best Study app NODIA

FIFTEEN MARKS QUESTIONS

67. Define a Java method temp-convert to convert Fahrenheit temperature to Celsius. The formula for the conversion is given as

$$C = \frac{5}{9}(F - 32), for\ F \text{ taken as } 45°F.$$

Ans : SQP 2024

```
public class Temp {
            public   static   double
tempConvert(float Fahren) {
          double cel = 5.0 / 9.0 *
(Fahren - 32);
        return cel;
    }

    public static void main() {
                          System.out.
println(tempConvert(45));
    }
}
```

68. Read the given segments of program and answer the questions that follow

```
    class Customer {
     private long Cust_id;
     String Cust_name;
     Customer(long id, String n) {
    Cust_id = id;
    Cust_name = n;
    }
public char funct() {
     System.out.println(Cust_id + " "
+ Cust_name);
     char chr = #;
     return chr;
}
}
```

(i) Write the access specifier of the class Customer

(ii) What is the scope of member Cust_id?

(iii) Name the members of the class that can be accessed in another class, namely Account

(iv) Name the members of the class that can be accessed in the main method contained in the same class Customer

(v) Define the main method that invokes the constructor and the member functions of the class

Ans : MAIN 2015

NODIA

(i) Public, by default

(ii) Cust_id has private access; its scope is limited to all functions of the class Customer

(iii) Member functions that can be accessed in another class

char funct()

Member variable (or Data members)

Cust_name

(iv) Cust_id, Cust_name, and function char funct() can be accessed in the main method of the same class (all member variables and member functions can be accessed in the same class)

(v)

```
public static void main() {
Customer obj = new Customer();
// Invokes the constructor
char d = obj.funct();
System.out.println(d);
}
```

69. Read the given segments of program and answer the questions that follow :

```
class Record {
public void method_Info(String
x) {
     char a = A, b = a;
     int c = a++ + ++b;
 }
}
```

(i) What value is stored in c after the method gets executed?

(ii) What is the value stored in a after the execution of the method?

(iii) Define the main method which calls the above method_Info

(iv) Modify method_Info so that the resultant c is displayed on the screen

Ans : SQP 2022

(i) ASCII code of A is 65, and of a is 97

In c = a++ + ++b;

a is post-incremented, so its original integer value 65 gets added to b (97)

Result is 65 + 97 = 162, i.e., c stores 162

(ii) as integer value is incremented by 1; it is 66 now, but the character variable a stores the character corresponding to ASCII code 66, i.e., B

(iii)

```
public static void main() {
Record obj = new Record();
obj.method_Info("#");
}
```

(iv)

```
public void method_Info(String
x){
char a = A, b = a;
int c = a++ + ++b;
System.out.println("Value of c
=" + c);
}
```

70. Read the given segments of program and answer the questions that follow :

```
class MyClass {
 short roll_no;
 long candidate_code;
 String cand_name;
}
```

(i) Define a member method of the class that displays the cand_name, candidate_code, and roll_no, the values being assigned as "ASTITVA", 2005123, and 20 respectively

(ii) Write the Java statement that calls the above-defined method

(iii) What happens if an object of the above class is created in another class Set_Account?

(iv) What happens if the statement short roll_no is changed to private short roll_no = 25;?

Ans : SQP 2023

(i)

```
class MyClass {
short roll_no;
long candidate_code;
String cand_name;

public void display() {
    roll_no = 20;
    candidate_code = 2005123;
    cand_name = "ASTITVA";
      System.out.println("NAME =
"    + cand_name);
      System.out.println("ROLL NO.
= " + roll_no);
      System.out.print("CODE = " +
candidate_code);
    }
}
```

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

(ii)

```
MyClass.display();
// or
MyClass ob = new MyClass();
ob.display();
```

(iii) Class MyClass has private access, so its object cannot be resolved in another class

(iv) Changing roll_nos visibility to private makes its scope limited to its own class, i.e., MyClass.

11. Read the given segments of program and answer the questions that follow :

```
public static int operate(int
                        a, int b) {
int c = a++ + ++a - a;
int d = c + b-- - --b;
return d;
```

(i) Write a single statement that calls the method operate if it is a member of class Mathematics

(ii) What happens if an object obct of class Mathematics directly invokes the variables c and d in another static method as follows

java

System.out.println(obct.c + obct.d);

(iii) What will be the final resultant values in c and d, if a and b are passed as 5 and 10 respectively?

Ans : COMP 2011

(i)

```
int x = Mathematics.operate(20,
10);
// or
Mathematics ob = new
Mathematics();
ob.operate(20, 10);
```

(ii) Variables d and c are local to method operate as they are defined and declared within the method body. If you call these in another static or non-static method, the compiler will not be able to resolve their location (generates error)

(iii) c stores 5 and d stores 23

Reasoning

```
c = a++ + ++a - a;
  = 5 + 7 - 7 = 5;

  d = c + b-- - --b;
  = 5 + 10 + 8 = 23;
```

12. Find errors in the following program segment and write the correct statements

1.

```
static sum(void) {
  int x, y;
  x = 2;
  y = +x;
system.out.println(x + " " + y);
}
```

2.

```
public static Void Main(string
                          x[]) {
int x, y;
y = sum(x);
System.out.println(y);
 }
```

3.

```
double product(int num1, num2) {
num1 * num2 = num3;
num2 * num3 = num4;
return num3, num4;
}
```

4.

```
public class Remainder {
private int a, b;
Remainder(int x, y) {
    a = x % y;
}
Remainder rem = new Remainder();
System.out.println(a);
}
```

5.

```
public class Name {
private String nam = "ABHISHEK";
public static void name(String
s[]) {
    Name obj = new Name();
    System.out.println(obj.nam);
}
}
```

6.

```
class CreateClass {
 private double a = 10;
 CreateClass(float m) {
     a = m;
 } /* end of constructor */
 void classFunction(int s) {
     int h = s + a;
     return h;
```

NODIA

```
    } /* end of classFunction */
     } /* end of CreateClass */

   class CheckAccess {
    public static void main() {
              CreateClass  obj  =  new
CreateClass(5.6);
        obj.a = 5;
        obj.classFunction(25);
    }
     }
```

7.

```
   public int checkError(void) {
    int x, y = 10;
    x = y++;
    System.out.println("Result = "
   + x + y);
}
```

8.

```
   private class Number {
    private double h;
   private void assignNumber(double
                                z) {
        h = 2;
    }
}

   public class checkError {
    checkError object = New
   checkError();
    checkError.assignNumber(15.26);
    System.out.println(h);
   }
   /* End of class checkError */
```

9.

```
   protected Class {
     long x, y;
     long Class() {
        x = 100;
        y = 100;
        }
    public static void main() {
        Class obj = new Class();
        x = x + 50;
        y = y + 55;
          System.out.println(obj.x +
obj.y);
    }
}
// End of main()
```

10.

```
String method1(String nam) {
    string x = nam();
}

public static void main() {
    String a = "India";
    string x = method1();
}
```

Ans : MAIN 2012

1. Error 1 : static sum(void)
   Correction : static int sum(void)
   Returning data type has not been specified in function definition
   Error 2 : int x, y;
   Correction : int x, y;
   Error 3 : y = +x;
   Correction : y = x;
   Error 4 : system.out.println(x + " " + y);
   Correction : System.out.println(x + " " + y);
2. Error 1 : public static Void Main(string x[ ])
   Correction : public static void main(String x[ ])
   Error 2 : int x, y;
   Correction : int z, y;
   z = 50; x cant be used as int variable as it is used as String type in main() as an argument
3. Error 1 : double product(int num1, num2)
   Correction : double product(int num1, int num2)
   Error 2 : num1 * num2 = num3;
   Correction : int num3 = num1 * num2;
   Error 3 : num2 * num3 = num4;
   Correction : int num4 = num2 * num3;
   Error 4 : return num3, num4;
   Correction : return num4;
   A function can return only one value
4. Error 1 : Remainder(int x, y)
   Correction : Remainder(int x, int y)
   Error 2 : Remainders object is created in the class without enclosing it in any function or constructor
   Error 3 : a is printed, but the statement is not again within a function/constructor body
   Error 4 : a is a non-static member that cannot be accessed directly in any static method like main()

Download INDIA's Best Study app NODIA

Downloaded from www.pdf.tube

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

Corrections :

```
public static void main(String
  args[]) {
   Remainder rem = new Remainder();
  System.out.println(rem.a);
       }
```

5. No syntax error

   But for every object of class Name, the string "ABHISHEK" will be stored in the member variable nam

6. Error 1 : void classFunction(int s)

   Correction : int classFunction(int s)

   As the function body contains a return statement that is returning an integer result h

   Error 2 : obj.a = 5;

   Correction : a is a private variable of class CreateClass and cant be accessed in class CheckAccess

7. Error 1 : public int checkError(void)

   Correction : public void checkError(void)

   Method body does not have a return statement

   Error 2 : System.out.println("Result = x, y");

   Correction : System.out.println("Result = " + x + y);

8. Error 1 : private void assignNumber(double 2)

   Correction : private void assignNumber(double a)

   Error 2 : System.out.println(h);

   h cannot be accessed in another class as it has private access in its class Number

   Error 3 : Class cannot be private

   Error 4 : Function assignNumber is a private member of class Number. It cannot be called in another class checkError

9. Error 1 : long Class() {

   Correction : Class() {

   Class is the constructor here; it cannot have a return type as long

   Error 2 : x = x + 50;

   Correction : obj.x = obj.x + 50;

   Non-static variable x cannot be referenced in a static method

   Error 3 : y = y + 55;

   Correction : obj.y = obj.y + 55;

   Error 4 : Class cannot be protected

   Correction : class Class

10. Error 1 : string x = nam();

    Correction : String x = nam;

    Error 2 : string x = method1();

    Correction : String x = method1(a);

    Error 3 : Function has a return type but no return statement

    Correction : return x;

73. (i) Define methods to find the square of a number and the cube of a number-the number must be passed to the methods from the calling statement, and the computed results must be returned to the calling module

(ii) Define a main method to call the above square and cube methods

Ans : COMP 2013

(i)

```
public class Q4a {
public static long square(int
                                num) {
     return num * num;
 }

 public static long cube1(int num)
{
     return num * num * num;
 }
}
```

(ii)

```
import java.util.*;
public class Q4b {
    public static void main() {
        Q4b obj = new Q4b();
      Scanner sc = new Scanner(System.
in);
          System.out.println("Please
enter a number : ");
        int n = sc.nextInt();

       System.out.println("1. Square
of number");
         System.out.println("2. Cube
of number");
           System.out.println("Enter
your choice : ");
        int ch = sc.nextInt();

        switch (ch) {
            case 1
```

NODIA

```
                long s = obj.square(n);
                System.out.println("The
square of " + n + " = " + s);
                   break;
              case 2
                long c = obj.cube1(n);
                System.out.println("The
cube of " + n + " = " + c);
                   break;
              default
                              System.out.
println("Invalid choice");
          }
     }
}
```

74. Define a method in Java Displacement that calculates the displacement covered by a moving body according to Newtons second law of motion

$$S = ut + \frac{1}{2}at^2$$

Where $u = 20m/s, t = 20\text{min}, a = 10m/s^2$

Ans : MAIN 2014

```
public class Q5 {
    static void displacement() {
         double u = 20, t = 20 * 60,
a = 10;
        double s = u * t + 1 / 2.0 *
a * t * t;
              System.out.println("The
displacement is : " + s + " metres");
    }

    public static void main() {
        displacement();
    }
}
```

75. Define a Java method to find the result of the following arithmetic expression $\frac{4.52a + b^2}{a - b}$ for the values of a and b taken as 2.2 and 3.5 , respectively

Ans : SQP 2015

```
public class Q6 {
  public static void expression(double
a, double b) {
        double r = (4.52 * a + b * b)
/ (a - b);
      System.out.println("The result
of the expression is : " + r);
    }
    public static void main() {
        double c = 2.2, d = 3.5;
        expression(c, d);
    }
}
```

Output: The result of the expression is : -17.072307692307692

76. (i) Write a method to print the result of the following function

$$f(x) = 2.4x^2 + 5.3x - 1.6,$$

For x taken as -2.

(ii) Write the main method to invoke the above-defined method.

Ans : COMP 2016

(i)

```
public static double f(double x)
{
return 2.4 * x * x + 5.3 * x - 1.6;
}
public static void main() {
double res = f(-2);
System.out.println("The result
                          is" + res);
}
```

Output : The result is -2.6

Note:

(i) Both methods f() and main() will be encapsulated within a class to get the output

(ii) Methodf()is static, so in the static main method, it is invoked directly without the use of any object as double res = f(– 2);

77. Write a complete Java program to print the ASCII codes of the following characters : ?, Z, m, #Define a method named ASC() to perform the above-stated job

Ans : MAIN 2017

```
public class Q8 {
    public static void ASC() {
        char c1 = ?;
        char c2 = Z;
        char c3 = m;
        char c4 = #;
        int ascii1 = c1;
        int ascii2 = c2;
        int ascii3 = c3;
        int ascii4 = c4;
```

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

```
        System.out.println("The ASCII
code of " + c1 + " character is = "
+ ascii1);
        System.out.println("The ASCII
code of " + c2 + " character is = "
+ ascii2);
        System.out.println("The ASCII
code of " + c3 + " character is = "
+ ascii3);
        System.out.println("The ASCII
code of " + c4 + " character is = "
+ ascii4);
    }

    public static void main(String[]
h) {
        ASC();
    }
}
```

Output
The ASCII code of ? character is = 63
The ASCII code of Z character is = 90
The ASCII code of m character is = 109
The ASCII code of # character is = 35

78. (i) Write a Java method result that takes the roll number, name, and marks of 3 subjects (engMarks, mathMarks, compMarks) as parameters from the mainThe method returns the percentage of the student, assuming that the maximum marks of each subject is 100

(ii) Define the main method for calling the result method

Ans : SQP 2018

(i)

```
public class Q9 {
   public double result(int roll,
   String name, int m1, int m2, int
   m3) {
        double t, per;
        t = m1 + m2 + m3;
        per = t / 300.0 * 100;
        return per;
    }
}
```

(ii)

```
   import java.util.*;
   public class Q9 {
    public static void main(String
   args[]) {
     Scanner sc = new Scanner(System.
in);
        System.out.println("Enter
Roll Number : ");
        int roll = sc.nextInt();

        System.out.println("Enter
Name : ");
        String name = sc.next();

        System.out.println("Enter
marks for English : ");
        int m1 = sc.nextInt();

        System.out.println("Enter
marks for Mathematics : ");
        int m2 = sc.nextInt();

        System.out.println("Enter
marks for Computer : ");
        int m3 = sc.nextInt();

        Q9 obj = new Q9();
          double percentage = obj.
result(roll, name, m1, m2, m3);
      System.out.println("Percentage
= " + percentage);
    }
}
```

79. (i) Define methods in the class Algebra to find the difference in the two algebraic expressions given as

$$f_1(x) = x^2 + 2x + 4$$

$$f_2(y) = -y^2 - 20y - 1$$

$$f_3(x,y) = f_1(x) - f_2(y)$$

Where values of x and y will be passed to the methods from main.

(ii) Define the main method to display the result of $f_3(x, y)$

Ans : COMP 2019

(i)

```
   import java.util.*;
   public class Algebra {
    static double f1(double x) {
        return x * x + 2 * x + 4;
    }

    static double f2(double y) {
       return -(y * y) - 20 * y - 1;
    }

   static double f3(double x, double
y) {
```

NODIA

```
        double result = f1(x) - f2(y);
         return result;
    }
}
```

(ii)

```
   public class Q10 {
    public static void main(String
 args[]) {
        Algebra ob = new Algebra();
        System.out.println("Please
 enter the two values");
     Scanner sc = new Scanner(System.
in);
        int r1 = sc.nextInt();
        int r2 = sc.nextInt();
            System.out.println("The
difference in two function values is
: " + ob.f3(r1, r2));
    }
}
```

80. Define the following methods in class Computation

(i) perimeter-rectangle : To calculate and return the perimeter of a rectangle

(ii) squarePara : To compute and return the perimeter of a square

(iii) main() : To call the above-defined methods

Ans : COMP 2020

```
import java.util.*;
public class Computation {
  static long perimeterRectangle(long
side1, long side2) {
        return 2 * (side1 + side2);
    }

   static long squarePara(long side)
{
        return 4 * side;
    }

     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
       System.out.println(" Menu ");
       System.out.println("1. To find
the rectangles perimeter");
       System.out.println("2. To find
the perimeter of square");
          System.out.println("Please
enter the choice 1/2 : ");
        int c = buf.nextInt();
        switch (c) {
            case 1
          System.out.println("Please
enter the two sides");
                    long s1 = buf.
nextLong();
                    long s2 = buf.
nextLong();
      long p = perimeterRectangle(s1,
s2);
             System.out.println("The
perimeter of the rectangle is : " +
p);
                break;

            case 2
                        System.out.
println("Please enter the side ");
             long s = buf.nextLong();
                      long para =
squarePara(s);
             System.out.println("The
perimeter of the square is : " +
para);
                break;

            default
                        System.out.
println("Invalid choice ");
        }
    }
}
```

81. Define a class Convert Currency that contains the following member methods

DollarToRupee() : Accepts dollars as a parameter and returns the amount in rupees

EuroToRupee() : Accepts euro currency as a parameter and returns the amount in rupees

main() : To call the above methods and display the result.

Ans : SQP 2021

```
public class ConvertCurrency {
   static double DollarToRupee(double
doll) {
        return doll * 65;
    }

    static double EuroToRupee(double
euro) {
        return euro * 44;
    }
```

NODIA

```
    public static void main(String
args[]) {
        System.out.println("Rupees
converted from 55 dollars : " +
DollarToRupee(55));
        System.out.println("Rupees
converted from 20 euros : " +
EuroToRupee(20));
    }
}
```

82. Write a class Triangle-area that contains the overloaded methods area, one that accepts the length of the base of a triangle and the height of the triangle as parameters. The other area method accepts the length of three sides of a triangle The two formulas for computing the area are

$$\text{area1} = \frac{1}{2} \times \text{base} \times \text{height}$$

$$\text{area2} = \sqrt{s(s-a)(s-b)(s-c)}$$

Ans : MAIN 2022

```
import java.util.*;
public class TriangleArea {
    static double area(int len, int
h) {
        return (1 / 2.0 * len * h);
    }

    static double area(int s1, int
s2, int s3) {
        double s = (s1 + s2 + s3) /
2.0;
      return (Math.sqrt(s * (s - s1)
* (s - s2) * (s - s3)));
    }

    public static void main(String
args[]) {
      System.out.println("Enter the
sides of a triangle:");
    Scanner inp = new Scanner(System.
in);
        int ss1 = inp.nextInt();
        int ss2 = inp.nextInt();
        int ss3 = inp.nextInt();
        System.out.println("The area
computed by Herons formula = " +
area(ss1, ss2, ss3));

        System.out.println("Enter
the base length and height of the
triangle:");
        int b = inp.nextInt();
        int height = inp.nextInt();
        System.out.println("The area
computed = " + area(b, height));
    }
}
```

83. Write a complete Java program that defines the following methods
assign() : Assigns value to a number
doublenumber() : Doubles the number assigned above.
display() : Displays the result
main() : Calls the above methods

Ans : COMP 2023

```
import java.util.*;
public class Q14 {
    private static double num, d;

    static void assign(double n) {
        num = n;
    }

    static void doublenumber() {
        d = num * 2;
    }

    static void display() {
            System.out.println("The
original number : " + num);
            System.out.println("The
doubled number : " + d);
    }

    public static void main(String
args[]) {
     Scanner sc = new Scanner(System.
in);
        System.out.println("Enter a
number : ");
        double n = sc.nextDouble();

        assign(n);
        doublenumber();
        display();
    }
}
```

Download INDIA's Best Study app NODIA

Downloaded from www.pdf.tube

84. Write a complete Java program that contains the following methods
assign() : Accepts values of the digit at units place, the digit at tens place, and the digit at hundreds place
findNumber() : Finds the number corresponding to the accepted digits
main() : Calls the above methods and displays the computed number

Ans : COMP 2024

```
import java.util.*;
public class Q15 {
     private static int unit, tens,
hun;

    static void assign(int u, int t,
int h) {
        unit = u;
        tens = t;
        hun = h;
    }

    static long findNumber() {
         return (hun * 100 + tens *
10 + unit);
    }

     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter a units place, the tens place
and hundreds place digits : ");
        int n1 = buf.nextInt();
        int n2 = buf.nextInt();
        int n3 = buf.nextInt();
        assign(n1, n2, n3);
       System.out.println("The number
is : " + findNumber());
    }
}
```

85. Design a class to overload a function compare() as follows
(i) void compare(int, int) - To compare two integer values and print the greater of the two integers
(ii) void compare (char, char) - To compare the numeric value of two characters and print the character with the higher numeric value
(iii) void compare (String, String) - To compare the length of two strings and print the longer of the two

Ans : MAIN 2000

```
public class Q16_OverloadCompare {
    void compare(int a, int b) {
        int c = 0;
        if (a > b) +{
            c = a;
        } else {
            c = b;
        }
         System.out.println("Greater
number is " + c);
    }

    void compare(char a, char b) {
        char c =  ;
        if (a > b)
            c = a;
        else
            c = b;
         System.out.println("Greater
character is " + c);
    }

    void compare(String a, String b)
{
        String c = " ";
        if (a.length() > b.length())
            c = a;
        else
            c = b;
          System.out.println("String
with greater length is " + c);
    }
}
```

Variables used
(i) a, b : Formal parameters used in the compare functions
(ii) c : Temporary local variable used in functions compare to store the higher value

86. Design a class to overload a function polygon area() as follows
(i) Double area(double a, double b, double c) With three double arguments, returns the area of a scalene triangle using the formula

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{Where } s = \frac{a+b+c}{2}$$

(ii) Double area(int a, int b, int height) With three integer arguments, returns the area of a trapezium using the formula

$$\text{area} = \frac{1}{2} \cdot \text{height} \cdot (a+b)$$

NODIA

(iii) Double area(double diagonal1, double diagonal2) With two double arguments, returns the area of a rhombus using the formula

$$\text{area} = \frac{1}{2} \cdot (\text{diagonal}\,1 \times \text{diagonal}\,2)$$

Ans : SQP 2001

```
The class is as follows
public class Overload {
     double area(double a, double b,
double c) {
        double s = (a + b + c) / 2.0;
        double areaOfTriangle = Math.
sqrt(s * (s - a) * (s - b) * (s - c));
         return areaOfTriangle;
    }

     double area(int a, int b, int
height) {
        double ar = 1.0 / 2 * height
* (a + b);
        return ar;
    }

      double area(double diagonal1,
double diagonal2) {
            double ar = 1.0 / 2 *
(diagonal1 * diagonal2);
        return ar;
    }
}
```

Variables used

a, b, c : Length of sides of a triangle
a, b, height : Length of two sides and the height of a trapezium
diagonal1, diagonal2 : Diagonals of a rhombus

87. Write a class with the name Volume using function overloading that computes the volume of a cube, a sphere, and a cuboid
Formula

$$\text{Volume of a cube } (V_c) = s*s*s$$

$$\text{Volume of a sphere } (V_s) = \frac{4}{3} \times \pi \times r \times r \times r$$

$$(\text{where } \pi = 3.14 \text{ or } \frac{22}{7})$$

$$\text{Volume of a cuboid (vcd)} = l \times b \times h$$

Ans : COMP 2002

```
class Volume {
    public static void vol(int S) /*
Taking side of cube as parameter of
function */ {
        int vc = S * S * S;
       System.out.println("Volume of
cube : " + vc);
    }

     public static void vol(double
r) /* Taking radius of sphere as
parameters */ {
        double vs = (4.0 / 3) * 3.14
* r * r * r;
       System.out.println("Volume of
sphere : " + vs);
    }

   public static void vol(int l, int
b, int h) /* Taking length, breadth,
and height as parameter */ {
        int vcd = l * b * h;
       System.out.println("Volume of
cuboid : " + vcd);
    }
}
```

Variables used :

S : Side of cube
l, b, h : Side of cuboid
r : Radius of sphere
vc : Volume of cube
vcd : Volume of cuboid
vs : Volume of sphere

88. Write a program using a function called area() to compute the area of:
(i) Circle πr^2 where $\pi = 3.14$
(ii) Square side × side
(iii) Rectangle *length* × *breadth*
Display the menu to output the area as per the users choice.

Ans : MAIN 2003

```
import java.util.*;

class Compute {
    public static void area() {
        System.out.println("Menu:");
         System.out.println("1. Area
of Circle");

         System.out.println("2. Area
of Square");

         System.out.println("3. Area
of Rectangle");

           System.out.println("Enter
choice:");
```

Click Here to Buy book on Amazon

```
      Scanner sc = new Scanner(System.
in);
        int y = sc.nextInt();

        switch (y) {
            case 1:
           System.out.println("Enter
radius:");
                int r = sc.nextInt();
                  double a1 = Math.PI
* r * r;
            System.out.println("Area
= " + a1);
                 break;

            case 2:
           System.out.println("Enter
side:");
                int d = sc.nextInt();
                 double a2 = d * d;
            System.out.println("Area
= " + a2);
                 break;

            case 3:
           System.out.println("Enter
length and breadth:");
                int l = sc.nextInt();
                int b = sc.nextInt();
                 int a3 = l * b;
            System.out.println("Area
= " + a3);
                 break;

            default:
                          System.out.
println("Invalid choice");
        }
    }
}
```

Variables used:

y : To store users choice

t : Radius of circle

d : Side of square

l, b : Length and breadth of rectangle

a1, a2, a3 : Area of circle, square, and rectangle respectively

Download INDIA's Best Study app NODIA

NODIA

89. Define a Java class Area that has the following members:

Members	Member Name	Description
Private variables	side1, side2, radius	Sides of rectangle, radius of circle
Two parameterized constructors	-	To assign values to sides and radius
Public method	rectArea()	Calculates area of rectangle
Public method	circArea()	Calculates area of circle
Public method	main()	Invokes all member methods and prints the results

Ans : SQP 2004

```
public class Area {
    private int side1, side2, radius;

    Area(int s1, int s2) {
        side1 = s1;
        side2 = s2;
    }

    Area(int r) {
        radius = r;
    }

    public int rectArea() {
        return side1 * side2;
    }

    public double circArea() {
       return 3.14 * radius * radius;
    }

    public static void main() {
        int rd = 5;
        Area ob1 = new Area(rd);

        int sd1 = 16, sd2 = 10;
      Area ob2 = new Area(sd1, sd2);

        int rect = ob2.rectArea();
         System.out.println("Area of
```

```
the rectangle = " + rect);

        double circ = ob1.circArea();
         System.out.println("Area of
the circle = " + circ);
    }
}
```

Output

Area of the rectangle = 160
Area of the circle = 78.5

90. Define a class named movie-magic with the following description:

Instance variables/data members:

int year – to store the year of release of a movie

String title – to store the title of the movie

float rating – to store the popularity rating of the movie (minimum rating = 0.0 and maximum rating = 5.0)

Member methods:

(i) movie-magic() : Default constructor to initialize numeric data members to 0 and String data member to ""

(ii) void accept() : To input and store year, title, and rating

(iii) void display() To display the title of a movie and a message based on the rating as per the table below

Rating	Message to be displayed
0.0 to 2.0	Flop
2.1 to 3.4	Semi-hit
3.5 to 4.5	Hit
4.6 to 5.0	Super Hit

Write a main method to create an object of the class and call the above member methods.

Ans : COMP 2004

The program code is as follows:

```
import java.util.Scanner;

public class movieMagic {
    int year;
    String title;
    float rating;

    movieMagic() {
        year = 0;
        title = "";
        rating = 0;
    }

    void accept() {
    Scanner scan = new Scanner(System.
in);

      System.out.print("Please enter
the year of release of the movie :
");
        year = scan.nextInt();

            System.out.print("Please
enter the title of the movie : ");
        title = scan.next();

            System.out.print("Please
enter the rating of the movie : ");
        rating = scan.nextFloat();
    }

    void display() {
         System.out.println("TITLE :
" + title);

         if (rating >= 0.0 && rating
<= 2.0) {
          System.out.println("Flop");
         } else if (rating >= 2.1 &&
rating <= 3.4) {
           System.out.println("Semi-
hit");
         } else if (rating >= 3.5 &&
rating <= 4.5) {
           System.out.println("Hit");
         } else if (rating >= 4.6 &&
rating <= 5.0) {
           System.out.println("Super
Hit");
        }
    }

    public static void main(String[]
args) {
             movieMagic obj = new
movieMagic();
        obj.accept();
        obj.display();
    }
}
if (rating >= 0.0 && rating <= 2.0)
System.out.println("Flop");
else if (rating >= 2.1 && rating <=
3.4) System.out.println("Semi-hit");
else if (rating >= 3.5 && rating <=
```

Downloaded from www.pdf.tube

```
4.5) System.out.println("Hit");
else if (rating >= 4.6 && rating
<= 5.0) System.out.println("Super
Hit");
}

public static void main(String[]
args) {
   movieMagic mov = new movieMagic();
    mov.accept();
    mov.display();
}
```

Variables used :

mov : object of the class

year : year of the release of the movie

title : title of the movie

rating : movie rating ranging from 0 to 5

91. Write a complete java program to increment the member variable values by 1. Define a method for the above task. Initialize the values for the variables through the constructor.

Ans : SQP 2005

```
public class Q22 {
    private static int i, j;

    Q22() {
        i = 5;
        j = 10;
    }

    public static void increment() {
        i++;
        j++;
    }

    public static void main(String[]
args) {
        increment();
         System.out.println("i=" + i
+ "\nj=" + j);
    }
}
```

Outputi=6j=11

92. Define a class called Parking-Lot with the following description. Instance variables/data members:

int vno – To store the vehicle number

int hours – To store the number of hours the vehicle is parked in the parking lot

double bill – To store the bill amount

Member methods:

(i) Void input() – To input the vno and hours

(ii) Void calculate() – To compute the parking charge at the rate of ₹3 for the first hour or part thereof, and ₹1.50 for each additional hour or part thereof

(iii) Void display() – To display the details

Write a main method to create an object of the class and call the above methods.

Ans : MAIN 2015

The program is as follows

```
import java.util.Scanner;

public class ParkingLot {
    int vno, hours;
    double bill;

    void input() {
    Scanner scan = new Scanner(System.
in);
         System.out.println("Please
enter vehicle number and hours");
        vno = scan.nextInt();
        hours = scan.nextInt();
    }

    void calculate() {
        if (hours <= 1) {
            bill = 3 * hours;
        } else {
              bill = 3 * 1 + 1.50 *
(hours - 1);
        }
    }

    void display() {
         System.out.println("VEHICLE
NO. is : " + vno);
        System.out.println("HOURS is
: " + hours);
         System.out.println("BILL is
:" + bill);
    }

    public static void main(String[]
args) {
     ParkingLot pl = new ParkingLot();
        pl.input();
        pl.calculate();
        pl.display();
    }
}
```

NODIA

Download INDIA's Best Study app NODIA

93. Write a complete Java program with the following members of its class Store Customer.

Members	Member Name	Description
private data members	Cust_Id	Stores character type customer id
	Cust_name	Stores customer name
	Cust_Add	Stores customer address
	Due_Amt	Stores the amount due to the customer
	Amount	Stores total to pay
public methods	getValues()	To store the data of a customer
	calculate()	To compute the total amount to be paid by the customer
	display()	To display all details

Ans : SQP 2007

```
import java.util.*;
public class StoreCustomer {
    private char cust_id;
     private String cust_name, cust_
add;
    double due_amt, amount;

     public void getValues(char cid,
String nam, String add, double due) {
         cust_id = cid;
         cust_name = nam;
         cust_add = add;
         due_amt = due;
    }

       public void calculate(double
present_amt) {
          amount = due_amt + present_
amt;
    }
    public void display() {
       System.out.println("*CUSTOMER
DETAILS*");
        System.out.println();

        System.out.println("CUSTOMER
IDENTITY CODE :" + cust_id);
       System.out.println("CUSTOMERS
NAME :" + cust_name);

         System.out.println("ADDRESS
:" + cust_add);
          System.out.println("AMOUNT
DUE :" + due_amt);

           System.out.println("TOTAL
AMOUNT :" + amount);
    }
}

/* Calling the methods defined in the
previous class */
class Call {
     public static void main(String
args[]) {

            StoreCustomer obj = new
StoreCustomer();
          System.out.println("Please
enter the customers id code, name,
and address:");
     Scanner sc = new Scanner(System.
in);

       char id = sc.next().charAt(0);
         String name = sc.next();
         String address = sc.next();

       System.out.println("Enter the
amount due:");
       double due = sc.nextDouble();

            obj.getValues(id, name,
address, due);
       System.out.println("Enter the
present amount paid:");

              double present_amt =
sc.nextDouble();

        obj.calculate(present_amt);
        obj.display();
    }
}
```

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

94. Given below is a hypothetical table showing rates of Income Tax for male citizens below the age of 65 years:

Taxable Income (TI) in ₹	Income Tax in ₹
Does not exceed ₹ 1,60,000	Nil
Is greater than ₹ 1,60,000 and less than or equal to ₹ 5,00,000	(TI - 1,60,000) × 10%
Is greater than ₹ 5,00,000 and less than or equal to ₹ 8,00,000	(TI - 5,00,000) × 20% + 34,000
Is greater than ₹ 8,00,000	(TI - 8,00,000) × 30% + 94,000

Write a program to input the age, gender (male or female), and Taxable Income of a person
If the age is more than 65 years or the gender is female, display "wrong category"
If the age is less than or equal to 65 years and the gender is male, compute and display the Income Tax payable as per the table given above.

Ans : COMP 2008

```
import java.util.Scanner;

public class Ans26 {
    public static void main(String[]
args) {
    Scanner scan = new Scanner(System.
in);
        double tax = 0;

          System.out.println("Please
Enter the age:");
        int age = scan.nextInt();

          System.out.println("Please
enter the gender (male/female):");
        String gender = scan.next();

          System.out.println("Please
enter the taxable income:");
       double ti = scan.nextDouble();

           if (age > 65 || gender.
equalsIgnoreCase("female")) {
           System.out.println("WRONG
CATEGORY");
      } else if (age <= 65 && gender.
equalsIgnoreCase("male")) {
            if (ti <= 160000) {
                tax = 0;
            } else if (ti > 160000 &&
ti <= 500000) {
                tax = (ti - 160000)
* 0.10;
            } else if (ti > 500000 &&
ti <= 800000) {
                tax = ((ti - 500000)
* 0.20) + 34000;
            } else {
                tax = ((ti - 800000)
* 0.30) + 94000;
            }
             System.out.println("TAX
IS  : ₹" + tax);
        }
    }
}
```

95. Define a class Units that contains:
(i) Private variables kg, g, km, m, l
(ii) Parameterized Constructor To initialize private variables kg, km, and l with the values that are passed to the constructor.
(iii) Method Convert To convert kg to g, km to m, and l to m^3, where kg stands for kilograms, km for kilometers, l for litres, and m for meters.

Ans : MAIN 2009

```
public class Units {
    private double kg, g, km, m, l;

      Units(double kilogram, double
gram, double met, double kilomet,
double litre) {
        kg = kilogram;
        g = gram;
        m = met;
        km = kilomet;
        l = litre;
    }

    void convert() {
        double cong = kg * 1000;
        double conmet = km * 1000;
        double conmL = (1000) * l;

          System.out.println(km + "
kilometers = " + conmet + " metres");
          System.out.println(kg + "
kilograms = " + cong + " grams");
           System.out.println(l + "
litres = " + conmL + " millilitres");
    }

     public static void main(String
args[]) {
```

Downloaded from www.pdf.tube

```
        Units obj = new Units(5, 2500,
500.5, 2.1, 2);
         obj.convert();
    }
}
```

96. Write a menu-driven program to perform the following : (Use a switch-case statement)

(i) To print the series 0, 3, 8, 15, 24, ... n term (value of n is to be an input by the user)

(ii) To find the sum of the series given below

$$S = \frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \frac{7}{8} + ... + \frac{19}{20}$$

Ans : SQP 2010

```
import java.util.*;

public class Q9Series {
    public static void series1() {
      Scanner sc = new Scanner(System.
in);
         System.out.println("Enter a
number:");
        int n = sc.nextInt();
        int d = 0;
       for (int i = 1; i <= n; i++) {
            d = i * i - 1;
          System.out.print(d + " ");
        }
        System.out.println();
    }

    public static void series2() {
        double sum = 0;

       for (int i = 1, j = 2; i < 20
&& j <= 20; i += 2, j += 2) {
            sum += (double) i / j;
        }
         System.out.println("The sum
of the series is " + sum);
    }

    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
        System.out.println("1. Print
the series");
         System.out.println("2. Find
the sum of the series");
           System.out.println("Enter
your choice:");
        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                series1();
                break;

            case 2:
                series2();
                break;

            default:
                        System.out.
println("Invalid choice");
        }
    }
}
```

97. Design a class to overload a function series() as follows :

(i) Void series(int x, int n) – To display the sum of the series given below

$$x^1 + x^2 + x^3 + ... + x^n \text{ terms.}$$

(ii) Void series(int p) – To display the following series 0, 7, 26, 63, ... p terms

(iii) void series() – To display the sum of the series given below

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ... \frac{1}{10}$$

Ans : COMP 2011

```
java
import java.util.Scanner;
class OverloadSeries {
    public static void series(int x,
int n) {
         double sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += Math.pow(x, i);
         }
          System.out.println("Sum of
the series : " + sum);
    }
    public static void series(int p)
{
         int x;
        for (int i = 1; i <= p; i++) {
            x = i * i * i - 1;
          System.out.print(x + " ");
        }
        System.out.println();
    }
    public static void series() {
        double sum = 0.0;
      for (int i = 2; i <= 10; i++) {
            sum += 1.0 / i;
        }
```

NODIA

```
        System.out.println("Sum is : " + sum);
    }

    public static void main(String[] args) {
            OverloadSeries obj = new OverloadSeries();
     Scanner sc = new Scanner(System.in);

          System.out.println("Choose the series to compute:");
       System.out.println("1. Sum of x^1 + x^2 + ... + x^n");
       System.out.println("2. Series : 0, 7, 26, 63, ...");
       System.out.println("3. Sum of the series 1/2 + 1/3 + ... + 1/10");
        int choice = sc.nextInt();

        switch (choice) {
            case 1:
           System.out.println("Enter the value of x and n:");
                int x = sc.nextInt();
                int n = sc.nextInt();
                 obj.series(x, n);
                 break;

            case 2:
           System.out.println("Enter the number of terms:");
                int p = sc.nextInt();
                 obj.series(p);
                 break;

            case 3:
                 obj.series();
                 break;

            default:
                          System.out.println("Invalid choice!");
        }
    }
}
```

Variables used:

x : base value for the first series.
n : number of terms in the first series.
p : number of terms in the second series.
sum : to store the sum of the series in each case.
choice : users menu selection.

98. Write a complete Java program that defines the following methods:

(i) assign() : Assigns a value to a string.
(ii) reverse-string() : Reverses the string assigned above.
(iii) display() : Displays the original and reversed strings.
(iv) main() : Calls the above methods.

Ans : COMP 2017

```
public class StringOperations {

    // Field to store the string
    private static String str;

     // Method to assign a value to the string
    public static void assign() {
        str = "Hello, Java!";
    }

    // Method to reverse the string
  public static String reversestring()
{
        String reversed = "";
       for (int i = str.length() - 1; i >= 0; i--) {
          reversed += str.charAt(i);
        }
        return reversed;
    }

    // Method to display the original and reversed strings
   public static void display(String reversedString) {
        System.out.println("Original String : " + str);
        System.out.println("Reversed String : " + reversedString);
    }
    // Main method to call the above methods
    public static void main(String[] args) {
        assign(); // Assign a string
            String reversedString = reversestring();  //  Reverse  the string
         display(reversedString); // Display the results
    }
}
```

www.nodia.in

Click Here to Buy book on Amazon

CHAPTER 4

CONSTRUCTORS

SUMMARY

1. CONSTRUCTOR

A constructor is a unique member function designed to initialize a class data members. Its name matches the class name, and it is automatically invoked whenever an object of the class is instantiated. Constructors ensure proper initialization without requiring explicit calls from the user.

1.1 General Format of Defining a Constructor

```
class-name()   // class-name is the name of the class
{
   statements;
}
```

1.2 Structure of a Class With Constructor

A Java class basically consists of :

(i) Member Variable Declarations : These store the state or attributes of the object

(ii) Used to initialize member variables, especially when private access restricts direct assignment from outside the class.

(iii) Methods : Define the behavior of the class.

The declared member variables in a class may have private access, which protects them from being used directly outside the class, i.e.,
objectName.variableDeclaredPrivate = Value;
is an invalid statement if written in another class. So, to assign values to such variables, constructors need to be defined.

1.3 Points to Remember While Defining Constructor

(i) Name of the constructor should be the same as that of the name of the class.

(ii) Do not write any returning data type like void, int, double, char, etc., before the constructor name.

(iii) A constructor should be defined with a public access specifier or no access specifier, in order to give it the default access.

(iv) Do not call the constructor explicitly. It gets called itself (i.e., the statements in the constructor get executed implicitly) when an object is created.

2. NON-PARAMETERIZED CONSTRUCTOR

A non-parameterized constructor does not accept any parameters or values from the object when it gets created.

3. DEFAULT CONSTRUCTOR

If there is no constructor defined in the class, and an object is created, it invokes the default constructor. It initializes default values to member variables.

3.1 Example of Default Constructor in My Class

```
class My Class {
      private int c;   /* Declaring member variable c */

    MyClass() {
      /* Defining a default constructor */
         c = 10;
    }
}
```

NODIA

4. PARAMETERIZED CONSTRUCTOR

These constructors have arguments. The values to these arguments are passed from the object when it is created.

4.1 Example of Parameterized Constructor

```
class MyClass2 {
    public MyClass2(int num) {
        c = num;
    }
}
```

5. CREATING AN OBJECT IN THE DEFAULT CONSTRUCTOR

```
class CreateObject {
    public static void main() {
        MyClass ob = new MyClass();
         /* Here MyClass is the name
of another class which is defined in
point 7 above, whose object is being
created here. */
    }
}
```

6. CREATING AN OBJECT IN PARAMETERIZED CONSTRUCTOR

```
class CreateObject2 {
    public static void main() {
        int a = 25;
       MyClass2 ob = new MyClass2();
       // Here, MyClass2 is the name
of another class and ob is its new
object.
    }
}
```

7. FORMAL PARAMETERS

The arguments in the first line of definition of the constructor are called formal parameters. Within the constructor definition, the values of these formal parameters (or formal variables) are used to initialize the private data members.

7.1 Example of Formal Parameters

```
class Numbers {
    int a, b;

    Numbers(int x, int y) {
        a = x;
        b = y;
    }
}
```

In the above example, x and y are formal parameters for the constructor.

8. ACTUAL PARAMETERS

The arguments passed to the constructor function from the object are called actual parameters. The values of actual parameters get copied to the formal parameters.

9. EXAMPLE OF ACTUAL PARAMETER

```
class AnotherClass {
    public static void main() {
        int p = 5, q = -1;
        Numbers obj = new Numbers(p,
q);
    }
}
```

In the above example, p and q are actual parameters which store actual values that are passed to formal parameters in class Numbers defined in point 14.

10. CONSTRUCTOR OVERLOADING

Defining more than one constructor for a class, where they differ in the parameter types or number of parameters passed to the constructor, is called constructor overloading. In this case, the constructors get called automatically with respect to the way an object is created.

11. EXAMPLE OF CONSTRUCTOR OVERLOADING

```
class Example {
    private int a, b; private float c;

    Example(int num1, int num2) {
        /* Constructor1 */
        a = num1;
        b = num2;
    }

    Example() {
        /* Constructor2 */
        a = 0;
        b = 0;
    }

    Example(float value, int num) {
        /* Constructor3 */
        c = value;
        a = num;
    }

    public static void main() {
        Example obj1 = new Example();
/* calls Constructor2 */
       Example obj2 = new Example(59,
20); /* calls Constructor3 */
        Example obj3 = new Example(5,
6);   /* calls Constructor1 */
    }
}
/* End of main */
/* End of class */
```

Download INDIA's Best Study app NODIA

Downloaded from www.pdf.tube

MULTIPLE CHOICE QUESTION

1. A special method that initializes values to data members is called _____.

(a) Constructor
(b) Method
(c) Function
(d) All of the above

Ans : MAIN 2024

(a) Constructor
A constructor is a special method in object-oriented programming used to initialize the values of data members when an object is created. It is automatically called when the object is instantiated.

2. Is the following statement true/false?
Access specifiers can be used for defining constructors.

(a) True
(b) False
(c) May be
(d) May not be

Ans : SQP 2023

(a) True
Access specifiers (public, private, and protected) can indeed be used for defining constructors in object-oriented programming. The access specifier determines the level of accessibility of the constructor

3. Which of the following is a return type of a constructor?

(a) Void
(b) Int
(c) Null
(d) None of the above

Ans : COMP 2021

(d) None of the above
A constructor does not have a return type, not even void. Constructors are special methods in object-oriented programming that are called automatically when an object is created. They initialize the object's properties but do not return any value.

4. If a constructor is declared as private, how many instances can be created for the class?

(a) 1
(b) 2
(c) Many
(d) 0

Ans : MAIN 2020

(a) 1
A private constructor restricts object creation, typically allowing only one instance via a factory method or singleton pattern.

5. How would the compiler/interpreter behave if there is no constructor defined in the class?

(a) No error. Default values will be initialized to members
(b) Error
(c) Impossible
(d) Program hangs

Ans : SQP 2018

(a) No error. Default values will be initialized to members
Java always provides a default constructor if no constructor is defined. Primitive types are initialized to their default values (0 for numeric types, false for boolean, null for references).

6. State the output of the below code:

```
class Abc {
    int a; String c;
    public static void main() {
        Abc ob = new Abc();
        System.out.print(ob.a + "  "
+ ob.c);
    }
}
```

(a) 1 * 1
(b) 0 0
(c) 0
(d) 0 null

Ans : COMP 2019

(d) 0 null
Instance variables of a class in Java are automatically initialized to their default values if not explicitly assigned. Default value for int is 0. Default value for String (a reference type) is null.

7. State the output of the below code:

```
class Abc {
    int a; String c;

    Abc(int x, String y) {
```

NODIA

```
        a = x;
        c = y;
    }

    public static void main() {
        Abc ob = new Abc(2, "!");
        System.out.print(ob.a + "  "
+ ob.c);
    }
}
```

(a) 2null
(b) error
(c) nullnull
(d) 2!

Ans : MAIN 2015

(d) 2!
The parameterized constructor initializes a and c with 2 and "!", respectively.

8. A constructor can only be written _____ in a class.
(a) At the beginning
(b) At the end
(c) Anywhere
(d) None of the above

Ans : SQP 2013

(c) anywhere
A constructor can be written anywhere within a class. Its placement in the code does not matter because it is identified by its name matching the class name and having no return type. The compiler recognizes it based on its signature.

9. An object is allocated the memory when the execution of _____ is completed.
(a) Main method
(b) Constructor
(c) Program
(d) All of the above

Ans : COMP 2012

(b) Constructor
An object is allocated memory when its constructor finishes execution. The constructor initializes the object, setting up its properties and resources, and once completed, the object is ready for use in memory.

Download INDIA's Best Study app NODIA

10. A constructor with no parameters is called _____.
(a) Non-parameterized constructor
(b) Parameterized constructor
(c) None
(d) Error

Ans : SQP 2010

(a) Non-parameterized constructor
A constructor with no parameters is called a non-parameterized or default constructor. It initializes the object with default values and does not require arguments during object creation.

11. A student tries to create a constructor in a class Sample. The constructor is written as:

```
void Sample() {
     System.out.println("Constructor
called");
}
```

What is incorrect in the above statement, and what will be the correct statement?
(a) Only A
(b) Only B
(c) All the three
(d) Both A and B

Ans : MAIN 2011

(a) Only A
The provided constructor is incorrectly defined as a method (with a return type). Constructors should not specify a return type.

12. What is the purpose of a constructor in a class?
(a) To define class methods
(b) To initialize class data members
(c) To destroy objects
(d) To handle exceptions

Ans : COMP 2015

(b) To initialize class data members
A constructor is specifically designed to initialize the data members of a class when an object is created.

13. Which statement about a constructor is TRUE?
(a) It must have a return type.
(b) Its name must match the class name.
(c) It must be explicitly called by the user.
(d) It can have a different name than the class.

Ans : SQP 2010

Click Here to Buy book on Amazon

(b) Its name must match the class name.
A constructor's name must exactly match the class name to ensure automatic invocation during object creation.

14. What is a non-parameterized constructor?
(a) A constructor that does not return anything.
(b) A constructor that does not accept parameters.
(c) A constructor with only default values.
(d) A constructor with a private access specifier.

Ans : COMP 2012

(b) A constructor that does not accept parameters.
Non-parameterized constructors do not take any arguments and usually set default values for data members.

15. What happens if no constructor is defined in a class?
(a) The program throws an error.
(b) A default constructor is automatically provided.
(c) The class cannot be instantiated.
(d) The object is created without initialization.

Ans : MAIN 2017

(b) A default constructor is automatically provided.
Java provides a default constructor that initializes member variables to default values if no constructor is explicitly defined.

16. Which access specifier is most commonly used for constructors?
(a) Private
(b) Public
(c) Protected
(d) Default

Ans : SQP 2019

(b) Public
Constructors are typically declared public to allow objects of the class to be created from any other class. If private, the class can only be instantiated from within the class itself, restricting object creation to controlled scenarios. Public access ensures the class can be widely used.

17. What is the primary function of a parameterized constructor?
(a) To return a value
(b) To set default values
(c) To accept arguments for initialization
(d) To invoke another constructor

Ans : MAIN 2021

(c) To accept arguments for initialization
Parameterized constructors are used to initialize class data members with specific values passed as arguments during object creation, offering flexibility and customization.

18. What is constructor overloading?
(a) Using multiple constructors with the same name but different access specifiers.
(b) Defining multiple constructors with different parameter lists.
(c) Calling one constructor from another.
(d) Overriding a constructor in a subclass.

Ans : COMP 2011

(b) Defining multiple constructors with different parameter lists.
Constructor overloading allows a class to have multiple constructors with varying numbers or types of parameters, providing different initialization options for objects.

NODIA

19. Which of the following is TRUE about formal parameters?
(a) They are values passed to a constructor during object creation.
(b) They are variables declared in the constructor definition.
(c) They are always initialized to default values.
(d) They are only used in default constructors.

Ans : SQP 2012

(b) They are variables declared in the constructor definition.
Formal parameters are the placeholders defined in the constructor's parameter list, receiving values passed by actual parameters during object creation.

20. What is the default value of a member variable in Java?
(a) null for all types
(b) 0 for numeric types, null for objects
(c) User-defined default values
(d) Compiler-defined random values

Ans : MAIN 2014

(b) 0 for numeric types, null for objects

In Java, member variables are automatically initialized with default values: 0 for numeric types, false for boolean, and null for object references, unless explicitly initialized.

21. Which of these correctly initializes an object using a default constructor?

(a) MyClass obj = new MyClass(5);

(b) MyClass obj = new MyClass();

(c) MyClass obj;

(d) obj = MyClass();

Ans : COMP 2019

(b) MyClass obj = new MyClass();
A default constructor takes no arguments. The syntax MyClass obj = new MyClass(); creates an object using the default constructor.

22. Which of these is an actual parameter?

(a) Variables declared in the constructor definition.

(b) Variables passed to a constructor during object creation.

(c) Parameters used to call a method.

(d) Variables that define the constructor.

Ans : SQP 2010

(b) Variables passed to a constructor during object creation.
Actual parameters are the values or variables passed to a constructor during object creation, which are then copied to formal parameters.

23. What is a formal parameter in a constructor?

(a) A variable declared outside the constructor.

(b) A placeholder variable declared in the constructor definition.

(c) A variable used to call a method.

(d) A parameter used to call another constructor.

Ans : MAIN 2015

(b) A placeholder variable declared in the constructor definition.
Formal parameters are variables defined in the constructor's parameter list and used to receive the values passed by actual parameters.

24. How do you define a parameterized constructor?

(a) By providing default values to all variables.

(b) By defining a constructor with parameters to accept values.

(c) By using a special access specifier.

(d) By defining a constructor without any parameters.

Ans : COMP 2017

(b) By defining a constructor with parameters to accept values.
A parameterized constructor accepts arguments that are used to initialize the data members of a class with specific values during object creation.

25. What happens when you call a constructor explicitly in Java?

(a) It initializes the object multiple times.

(b) It throws an error.

(c) The constructor cannot be explicitly called.

(d) It overrides the default constructor.

Ans : MAIN 2019

(c) The constructor cannot be explicitly called.
In Java, constructors are automatically invoked during object creation and cannot be explicitly called like methods.

26. Which of these is an example of constructor overloading?

(a) A class having multiple constructors with the same name and parameter lists.

(b) A class having multiple constructors with different parameter lists.

(c) A class having a single constructor.

(d) A class having private constructors only.

Ans : COMP 2021

(b) A class having multiple constructors with different parameter lists.
Constructor overloading refers to having multiple constructors in a class, differing in the number or types of parameters.

27. Which constructor is automatically invoked if none is defined in the class?

(a) Parameterized constructor

(b) Default constructor

(c) Private constructor

(d) Static constructor

Ans : SQP 2000

(b) Default constructor
If no constructor is defined in a class, the compiler provides a default constructor that initializes the object with default values.

NODIA

Download INDIA's Best Study app NODIA

28. Which of these is an invalid constructor definition?

(a) public MyClass() { }

(b) void MyClass() { }

(c) private MyClass() { }

(d) MyClass() { }

Ans : MAIN 2001

(b) void MyClass() { }
Constructors cannot have return types, even void. Including a return type makes it a regular method, not a constructor.

29. Which access specifier allows the constructor to be called only within its class?

(a) public

(b) private

(c) protected

(d) default

Ans : COMP 2010

(b) private
A private constructor restricts object creation to within the class itself, preventing access from outside the class.

30. In the example Example(float value, int num), what is value?

(a) A member variable

(b) An actual parameter

(c) A formal parameter

(d) A static variable

Ans : MAIN 2011

(c) A formal parameter
In the constructor definition, value is a formal parameter used to receive values passed during object creation.

31. What will happen if a constructor is declared private in a class?

(a) It cannot be accessed outside the class.

(b) It initializes default values.

(c) It can be called only by public methods.

(d) It throws an error during compilation.

Ans : SQP 2013

(a) It cannot be accessed outside the class.
A private constructor limits object creation to within the class, often used in singleton design patterns.

Download INDIA's Best Study app NODIA

32. A student writes the following code to initialize a member variable num to 10 using a default constructor:

```
class Demo {
    private int num;
    Demo(int value) {
        num = value;
    }
}
```

Which is the correct way to define the default constructor in this class?

A. Demo() { num = 10; }

B. Demo { num = 10; }

C. public Demo() { num = 10; }

(a) Only A

(b) Only C

(c) Both A and C

(d) All the three

Ans : COMP 2008

(c) Both A and C
The correct way to define a default constructor in Java is by using the class name followed by an empty parameter list, optionally including the public access modifier. In the given options, A (Demo() { num = 10; }) is valid syntax but does not use the public modifier, which is considered a best practice though not mandatory. Option B (Demo { num = 10; }) is incorrect because constructors in Java must have parentheses, even if they do not take any parameters. Option C (public Demo() { num = 10; }) is the proper and most conventional way to define a default constructor.

33. A student tries to create an object of a class Example with a parameterized constructor using the following code:

```
Example obj = new Example();
```

Which is the correct way to create the object if the class constructor is defined as:

```
Example(int num) { }
```

A. Example obj = new Example(5);

B. Example obj = new Example();

C. Example obj = Example(5);

(a) Only A

(b) Only B

(c) Both A and C

(d) All the three

Ans : SQP 2007

Downloaded from www.pdf.tube

(a) Only A
In Java, if a class has only a parametrized constructor (e.g., Example(int num)), the default no-argument constructor is not available. To create an object, you must pass the required parameter to the constructor. Thus, Example obj = new Example(5); is the correct way to instantiate the object. Options B and C are invalid due to syntax errors or missing arguments.

34. A student defines two constructors for a class Test, as follows:

```
class Test {
    Test() { }
    void Test(int x) { }
}
```

A. Both are constructors.
B. Only the first one is a constructor.
C. Both are methods.
(a) Only A
(b) Only B
(c) Only C
(d) Both A and C

Ans : MAIN 2005

(b) Only B
The first definition, Test(), is a constructor because it has no return type and its name matches the class name. The second definition, void Test(int x), is not a constructor because it has a return type (void). It is a method, not a constructor. Constructors in Java cannot have return types.

35. A student writes the following code to overload a constructor in a class Shape:

```
class Shape {
    Shape(int length) { }
    Shape(int length, int width) { }
    Shape(double radius) { }
}
```

Which statement is true about constructor overloading?
(i) Overloading works because of different numbers of parameters.
(ii) Overloading works because of different parameter types.
(iii) Overloading does not work in constructors.
(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans : SQP 2003

(c) Both A and B
Constructor overloading in Java allows multiple constructors in a class to have the same name (e.g., Shape) as long as they differ in number of parameters (option A) or parameter types (option B). In the given code, the constructors Shape(int length), Shape(int length, int width), and Shape(double radius) demonstrate these variations. Option C is incorrect because constructor overloading is valid and commonly used in Java.

36. Consider the following program segment where statements are jumbled. Choose the correct order to define a default constructor and initialize a variable.

```
class Demo {
    private int num;

    Demo() {
        System.out.println("Default
constructor called"); // (1)
        num = 0; // (2)
    }
}
```

(a) (1) (2)
(b) (2) (1)
(c) Both orders are valid
(d) Neither order is valid

Ans : COMP 2002

(c) Both orders are valid.
In a constructor, the order of statements does not affect the functionality as long as they are logically valid and follow the language syntax. In this case, initializing the variable num and printing the message using System.out.println can happen in any order without affecting the program's correctness. Both (1) (2) and (2) (1) are valid sequences.

37. Consider the following jumbled code segment to swap two numbers without using a third variable. Choose the correct order of statements.

```
void swap(int a, int b) {
    a = a + b;  // (1)
    b = a - b;  // (2)
    a = a - b;  // (3)
}
```

(a) (1) (2) (3)
(b) (3) (2) (1)
(c) (2) (1) (3)
(d) (3) (1) (2)

NODIA

Ans : MAIN 2000

(a) (1) (2) (3)

The steps to swap two numbers a and b without using a third variable are as follows:

(i) Add a and b and store the result in a: a = a + b.

(ii) Subtract the new value of a (which is a + b) by b and store the result in b: b = a - b.

(iii) Subtract the new value of b from a to assign the original value of b to a: a = a - b.

38. Consider the following jumbled statements for creating an object using a parameterized constructor. Choose the correct order of statements.

```
int value = 10;                    // (1)
MyClass obj = new MyClass(value);// (2)
class MyClass {                    // (3)
    MyClass(int num) {             // (4)
     System.out.println("Constructor called with value: " + num);
    }
}
```

(a) (3) (4) (1) (2)

(b) (1) (3) (4) (2)

(c) (3) (1) (4) (2)

(d) (4) (3) (1) (2)

Ans : SQP 2001

(a) (3) (4) (1) (2)

To create an object using a parameterized constructor, the correct sequence is as follows:

(i) Define the class: class MyClass { ... } (3).

(ii) Define the parameterized constructor: MyClass(int num) { ... } (4).

(iii) Initialize a variable to pass to the constructor: int value = 10; (1).

(iv) Create the object using the parameterized constructor: MyClass obj = new MyClass(value); (2).

39. Consider the following program segment to define and use a constructor. The statements are jumbled. Choose the correct order of statements.

```
class Test {                       // (1)
    Test(int a) {                  // (2)
        System.out.println("Value: " + a);  // (3)
    }
    public static void main(String[] args) { // (4)
            Test obj = new Test(5);
// (5)
    }
}
```

(a) (1) (2) (3) (4) (5)

(b) (1) (4) (2) (5) (3)

(c) (1) (4) (5) (2) (3)

(d) (1) (2) (4) (5) (3)

Ans : COMP 2002

(a) (1) (2) (3) (4) (5)

To define and use a constructor correctly, the sequence of statements must follow these steps:

(i) Define the class: class Test { (1).

(ii) Define the constructor: Test(int a) { (2).

(iii) Add the logic inside the constructor: System.out.println("Value: " + a); (3).

(iv) Define the main method: public static void main(String [ ] args) { (4).

(v) Create the object and call the constructor: Test obj = new Test(5); (5).

40. **Assertion (A) :** A constructor must not have a return type in Java.

Reason (R) : If a return type is specified, the method will be treated as a regular method, not a constructor.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : SQP 2003

(a) Both A and R are true, and R is the correct explanation of A.

In Java, a constructor must not have a return type, including void. If a return type is specified, the method is treated as a regular method instead of a constructor. This is because constructors are special methods used to initialize objects, and their unique behavior is defined by the absence of a return type. Therefore, the assertion that "a constructor must not have a return type" is true, and the reason provided that specifying a return type makes it a regular method is also true and serves as the correct explanation.

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

41. **Assertion (A) :** Default constructors are automatically provided in a class if no other constructors are explicitly defined.
Reason (R) : The default constructor initializes member variables to their default values, like 0 for integers and null for objects.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2004

(a) Both A and R are true, and R is the correct explanation of A.
In Java, if no constructors are explicitly defined in a class, the compiler automatically provides a default constructor. This default constructor initializes the member variables to their default values, such as 0 for numeric types, false for boolean, and null for reference types. The reason (R) accurately explains why the assertion (A) is true, as the purpose of the default constructor is to ensure member variables have valid initial values.

42. **Assertion (A) :** Constructor overloading in Java is possible by defining multiple constructors with the same name but different parameter lists.
Reason (R) : Constructor overloading works due to the compile-time resolution of parameter types and counts.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2005

(a) Both A and R are true, and R is the correct explanation of A.
Constructor overloading is allowed in Java, and the compiler differentiates constructors based on their parameter types and counts.
explanation-Overloading works due to compile-time resolution of different parameter types/ counts.

43. **Assertion (A) :** A parameterized constructor allows passing values to initialize object attributes during object creation.
Reason (R) : Formal parameters in the constructor definition are used to initialize private member variables.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : SQP 2006

(a) Both A and R are true, and R is the correct explanation of A.
A parameterized constructor is designed to allow the passing of values during object creation, which can then be used to initialize the attributes of the object. Formal parameters in the constructor serve as placeholders for these values and are often used to initialize private member variables, ensuring encapsulation. Therefore, both the assertion and the reason are true, and the reason correctly explains the assertion.

44. **Assertion (A) :** Non-parameterized constructors are also known as default constructors in Java.
Reason (R) : Non-parameterized constructors take no arguments and assign custom or default values to member variables.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2007

(a) Both A and R are true, and R is the correct explanation of A.
Non-parameterized constructors, also known as default constructors, are constructors that take no arguments. They can either be explicitly defined by the programmer or implicitly provided by the compiler if no other constructors are present. These constructors typically initialize member variables to default or custom values. Hence, both the assertion and the reason are true, and the reason provides a correct explanation of the assertion.

NODIA
Download INDIA's Best Study app NODIA

(iv) What will be the output if the above code is executed assuming that method func() is called in the main method?

Ans : COMP 2005

(i) GetOutput

(ii) a will store 4 at the time of object creation, when the constructor gets invoked.

(iii) After the method GetOutput gets executed, a = 4.

(iv) 9

68. Read the following code segments and answer the questions that follow:

```
Private boolean var;
private int value;

public Sample(boolean m, int var) {
    var = m;
    value = 2 * var - --var;
}

public void func() {
    int k = 4; boolean b = false;
    Sample s = new Sample(b, k);
}
```

Questions :

(i) Name the formal parameters in the code.

(ii) Name the actual parameters in the code.

(iii) Will the Sample constructor get invoked if an object of the same class is created in another class My World ? Give reason.

(iv) What is the result stored in value?

Ans : MAIN 2006

(i) Sample

(ii) m and var are the formal parameters.

(iii) b and k are the actual parameters.

(iv) Yes, the constructor will get invoked because it has been specified with public access.

(v) Value stores 10.

$$2 \times 4 + 2 = 10$$

(var is decremented by 1 twice).

69. Read the following code segments and answer the questions that follow:

```
class Sum {
    private int x, y;
    public int total;

    public Sum(int a, int b) {
        x = a;
        y = b;
        total = x + y;
    }

    public static void main(String
x[]) {
        Sum tot = new Sum(50, 46);
       System.out.print("Total of 50
and 46 is " + tot.total);
    }
}
```

Questions :

(i) Write all data members of class Sum.

(ii) Write all member methods of the class Sum.

(iii) Can you access variables x and y in its public static void main() as follows?

```
System.out.print("Total of " +
tot.x + " and " + tot.y + "=" +
tot.total);
```

Give reason for your answer.

(iv) Can you access x and y in another class MyAccounts public static void main() as follows?

```
System.out.print("x is " + x);
System.out.print("y is " + y);
```

Write the statement if there is any other way to access it.

(v) In the above code, how many times will Sum() get invoked?

Ans : SQP 2007

(i) Data members of class Sum: x, y, total.

(ii) Member methods: Sum (constructor) and main().

(iii) Yes, variables x and y are accessible in public static void main() as given, even though they are declared private. Here, main() is defined in the same class of which x and y are members. Private members are accessible in all member methods of the same class. So, the given statement is valid.

(iv) Theres no way by which x and y can be accessed in any other class till they are declared as private. If the access specifier of x and y is changed to public, then they can be accessed in the public static void main() method of MyAccount class in the following way:

```
class MyAccount {
public static void
main(String 1x[]) {
MyAccount obj = new MyAccount();
obj.x = 10;
```

Downloaded from www.pdf.tube

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

```
for (int i = 1; i <= num; i++) {
    factorial *= i;
}

System.out.println(factorial);
```

Ans : SQP 2013

Error : Logical Error

```
String value = "5";

int num = Integer.parseInt(value);

int factorial = 1;

for (int i = 1; i <= num; i++) {
    factorial *= i;
}

System.out.println(factorial);
```

52. A student executes the following program segment and gets an error. Identify the statement that has an error and correct it to get the output as "PASS".

```
char grade = 'A';

switch (grade) {
  case 1: System.out.println("PASS");
break;
  case 2: System.out.println("FAIL");
break;
}
```

Ans : COMP 2014

Error : The switch statement cases (case 1, case 2) must match the type of the switch expression (char grade).

Correction : Replace the case values with valid char values that correspond to the grade variable.

```
char grade = 'A';

switch (grade) {
            case   'A':   System.out.
println("PASS"); break;
            case   'F':   System.out.
println("FAIL"); break;
}
```

53. Explain the concept of constructor overloading.

Ans : SQP 2015

Constructor Overloading : Defining more than one constructor for a class where they differ in the parameter types or number of parameters passed to the constructor is called constructor overloading. In this case, the constructors get called automatically with respect to the way an object is created.

54. State two features of a constructor.

Ans : COMP 2016

Features of constructor:

(i) The name of the constructor is same as the class name.

(ii) A constructor automatically gets invoked whenever an object of the class is created.

55. What is a default constructor?

Ans : MAIN 2017

A default constructor initializes default values to the member variables. Sometimes, a non-parameterized constructor is also referred to as a default constructor. A default constructor does not accept any parameter or values from the object when it gets created.

Example of default constructor in MyClass :

```
class MyClass {
    private int c;

    MyClass() {
        c = 10;
    }
}
```

56. Enter any two variables through constructor parameters and write a program to swap and print the values.

Ans : SQP 2018

```
class Swap {
    int a, b;
    Swap(int x, int y) {
        a = x;
        b = y;
    }
    void swapValues() {
        int temp = a;
        a = b;
        b = temp;
    }
    void display() {
         System.out.println("a = " +
a + ", b = " + b);
    }

    public static void main() {
        Swap obj = new Swap(10, 20);
        obj.swapValues();
```

NODIA

```
        obj.display();
    }
}
```

57. Define a parameterized constructor for the class Floppy that assigns the diameter and storage space passed to it to the member variables diameter and space.

Ans : COMP 2019

```
class Floppy {
    double diameter, space;

    Floppy(double d, double s) {
        diameter = d;
        space = s;
    }
}
```

58. Write the difference between default constructor and parameterized constructor.

Ans : MAIN 2020

(i) A default or non-parameterized constructor does not accept any values as parameters.

(ii) A parameterized constructor accepts arguments. The initial values are supplied to the arguments at the time of object creation.

59. Can the number of actual parameters be different from the number of formal parameters?

Ans : SQP 2021

No, the number of actual parameters should be the same as the number of formal parameters defined in the constructor if the object is attempting to invoke the same constructor.

60. If a programmer does not define a constructor in a class, cant the programmer able to create an object in the program?

Ans : COMP 2022

If a constructor is not defined, the programmer can very well create an object of the class because the compiler itself supplies a default constructor that initializes all instance variables (member variables) to default values.

61. Create a class with one integer instance variable. Initialize the variable using:

(i) Default constructor

(ii) Parameterized constructor

Ans : MAIN 2023

```
class Test {
    int a;

    Test() {
         // variable initialization
using default constructor
        a = 0;
    }

    Test(int x) {
         // variable initialization
using parameterized constructor
        a = x;
    }
}
```

62. Write the difference between two constructors, if one constructor is defined with the private/ protected access and the other one is defined with the public access?

Ans : MAIN 2000

If a constructor is defined with a public access, the object of the class can invoke the constructor irrespective of the class where the object is being created. But, if the constructor is defined with a private or protected access, the constructor of the class cannot be invoked during object creation in another class. It can get called only if the object is created in a method belonging to the same class or child class.

63. Read the following code segments and answer the questions that follow:

```
class Abc {
    static int j, i;

    Abc() {
      System.out.println("Abc object
is now created");
    }  /* Constructor1 */
    Abc(int a) {
        i = a;
        j *= i;
    }  /* Constructor2 */
}
```

Questions :

(i) What are the values of j and i before any object is created?

(ii) Write the statements to create objects that invoke constructors 1 and 2.

(iii) Compute the values of i and j if the actual value passed to Abc(int a) is 5.

Click Here to Buy book on Amazon

Ans : SQP 2001

(i) j = 0 and i = 0 before any object is created.

(ii)

```
class Abc {
public static void main() {
  Abc object1 = new Abc();
/*Invokes Constructor1 */
  Abc object2 = new Abc(25);
/* Invokes Constructor2 */
   }
}
```

(iii) i = 5 and j = 0 [as j* = i is equivalent to (j = j*i = 0*5 = 0

64. Why do constructors need to be overloaded?

Ans : COMP 2002

Constructor overloading allows different initial values for different objects. For example, a default constructor may initialize private, integer variables a and b as 0. Another constructor may accept int values x and y to be assigned to a and b. Yet, another constructor may accept two integer and one float value to be assigned to a, b, and a float variable c.

These constructors get invoked depending on the parameters passed to constructors during object creation.

Example :

```
class Example {
    int a, b;
    double c;

     Example() { a = 0; b = 0; }
/* Constructor1 */
    Example(int x, int y) { a = x; b
= y; }  /* Constructor2 */
   Example(int x, int y, double z) {
        a = x; b = y; c = z;
    }

     public static void main(String
x[]) {
        Example obj = new Example(5,
6, 2.5); /* Invoking Constructor3 */
       Example obj1 = new Example();
/* Invoking Constructor1 */
        Example st = new Example(30,
10);     /* Invoking Constructor2 */
    }
}
```

Download INDIA's Best Study app NODIA

65. State the difference between Constructor and Method.

Ans : MAIN 2016

Constructor :

(i) A constructor is a member of the class whose name is exactly same as the class name.

(ii) It has no returning data type not even void.

(iii) It gets automatically invoked whenever an object is created.

(iv) Purpose of writing a constructor is to initialize values to data members.

Method :

(i) A method is also a member of the class but with a name different from the class name.

(ii) It must have a returning data type. It can be void, if it doesnt return any value.

(iii) It gets invoked explicitly.

(iv) A method is written to perform some process/ function on data.

66. What is the garbage collection feature of Java?

Ans : SQP 2004

In Java, the new operator allocates the memory for the objects during the runtime of a program. The memory occupied by these objects is deallocated when no references to the objects exist. This memory can be reclaimed. In Java, this deallocation is performed automatically and is called the garbage collection feature.

67. Read the following code segments and answer the questions that follow :

```
private int x;
   public GetOutput(short a) {
    x = ++a + --a;
    System.out.println(x);
     }

   public void func() {
    GetOutput obj = new GetOutput(4);
     /* Assume main method calls
         func()*/
     }
```

Questions :

(i) Name the class in which the above code is defined.

(ii) What value will the variable a store when the object is generated (created)?

(iii) What value will a store after the code gets executed?

Downloaded from www.pdf.tube

```
                Salesman obj1 = new
Salesman(15000, 2);
          double ComInRupees = obj1.
commission / 100 * obj1.sales;
      System.out.println("Commission
in Rupees = ₹" + ComInRupees);
    }
}
```

Variables used :
commission, sales : Instance variables to store commission in percent and sales in rupees.
s, c : Formal parameters of constructor.
ComInRupees : Stores commission in Rupees.

78. Define a class named Fruit Juice with the following description :
Instance variables/data members:
(i) int product_code – stores the product code number
(ii) String flavour – stores the flavour of the juice (e.g., orange, apple, etc.)
(iii) String pack_type – stores the type of packaging (e.g., tetra-pack, PET bottle, etc.)
(iv) int pack_size – stores the package size (e.g., 200 ml, 400 ml, etc.)
(v) int product_price – stores the price of the product
Member methods :
(i) FruitJuice() – Default constructor to initialize integer data members to 0 and String data members to "".
(ii) void input() – To input and store the product code, flavour, pack type, pack size, and product price.
(iii) void discount() – To reduce the product price by 10.
(iv) void display() – To display the product code, flavour, pack type, pack size, and product price.

Ans : MAIN 2016

The class is as follows :

```
import java.util.Scanner;
public class FruitJuice {
    int product_code;
    String flavor;
    String pack_type;
    int pack_size;
    int product_price;

    public FruitJuice() {
        product_code = 0;
        flavor = "";
        pack_type = "";
        pack_size = 0;
        product_price = 0;
    }

    void input() {
    Scanner scan = new Scanner(System.
in);
          System.out.println("Enter
the product code, flavour, and pack
type");
      product_code = scan.nextInt();
        flavor = scan.next();
        pack_type = scan.next();
       System.out.println("Enter the
pack size and product price");
        pack_size = scan.nextInt();
     product_price = scan.nextInt();
    }

    void discount() {
       product_price = product_price
- 10;
    }

    void display() {
         System.out.println("Product
Code: " + product_code);
        System.out.println("Flavour:
" + flavor);
            System.out.println("Pack
Type: " + pack_type);
            System.out.println("Pack
Size: " + pack_size + " ml");
         System.out.println("Product
Price: ₹" + product_price);
    }
}
void display() {
    System.out.println("Product code
is: " + product_code);
     System.out.println("Flavour is:
" + flavor);
    System.out.println("Pack type is:
" + pack_type);
    System.out.println("Pack size is:
" + pack_size);
   System.out.println("Product Price
is: " + product_price);
}
Variables used:
product_code: Code of the product
pack_size: Package size in ml
```

NODIA

```
        obj.y = 20;
    System.out.println("x is " + obj.x);
    System.out.println("y is " + obj.y);
```

(v) Sum constructor will get invoked only once, as only one object tot is created in the given code.

70. Read the following code segments and answer the questions that follow:

```
class Display {
  private short a; private int b;

  public static void main(String
  x[]) {
  System.out.println(20 + "*" + 30
  + "=" + 20 * 30);
  System.out.println(20 + "*" + 31
  + "=" + 20 * 31);
   }
   }
```

Questions :

(i) How many times will the compiler call the constructor? Give reason.

(ii) What are the values of a and b, if not assigned explicitly?

(iii) Write the output of the above code.

Ans : COMP 2008

(i) The constructor will not be called by the compiler even once because no object is being created.

(ii) a and b both will store zero.

(iii) $20 \times 30 = 600$

$20 \times 31 = 620$

71. Read the following code segments and answer the questions that follow:

```
class Code {
   private int c;
   public int d;

    Code(int i) { d = i; }  /* Line
No.1 */
     Code(Code A) { c = A.d; }  /*
Line No.2 */
}
```

Questions :

(i) Explain line No.1 in the above code.

(ii) What is the data type of the formal parameter in line No.2?

(iii) What is the name given to the constructor defined in line No.2 and line No.1?

(iv) Explain line No.2.

(v) Write the main method in a class Employee to call the constructors defined in class Code.

Ans : MAIN 2009

(i) Line No. 1 in the given code gives the definition of a parameterized constructor that accepts an integer value and assigns it to the integer member variable.

(ii) The data type of the formal parameter A in line No. 2 is the class type. Thus, an object of class Code will be passed to the constructor.

(iii) Parameterized Constructors

(iv) Line No. 2 gives the constructor definition which accepts an object of class Code as its parameter.

(v)

```
 class Employee {
public static void main(String x[])
{
Code obj = new Code(6);
System.out.print(obj.d);
}
}
```

72. Distinguish between the following pairs of statements:

(i)

```
Time() { }
Time() { a = 0; b = 0; }
```

(ii)

```
Date dt = new Date(20, 08, 2000);
Date dt = new Date();
```

Ans : SQP 2010

(i) If a Time class contains the constructor defined as Time() { }, the constructor gets invoked on creation of an object but does nothing while the second constructor when invoked would initialize data members a and b to value zero.

(ii) The first statement creates an object dt by invoking its parameterized constructor. The object gets initialized with values 20, 8, and 2000. While the second statement creates an object and calls the default constructor.

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

TWO MARKS QUESTIONS

45. (i) How many constructors can a class have?
(ii) Why do constructors not have any return type?

Ans : SQP 2024

(i) A class can have any number of constructors defined for itself provided that they differ in their arguments.

(ii) Constructors do not have any return type, not even void, because the implicit return type of a constructor is the class type itself.

46. Which unit of a class gets called when the object of the class is created?

Ans : COMP 2008

A constructor gets called when the object of the class is created.

47. What is the use of a constructor in a Java class?

Ans : SQP 2009

A constructor is generally used to assign the initial values to private data members of the class. The private data members are not accessible outside the class by any means. And if there arises a need to initialize them with values for an object, then they can be assigned in a constructor. This constructor gets invoked automatically when an object of the class is created in the same or any other class.

48. What are the two types of constructors? [2015]

Ans : MAIN 2010

The two types of constructors are:

(i) **Default Constructor :** A constructor that takes no arguments and initializes the object's attributes with default values.

(ii) **Parameterized Constructor :** A constructor that takes arguments to initialize the object's attributes with specific values provided during object creation.

49. A student is trying to convert a string present in s to an integer value so that they can calculate the cube of the converted value. However, the code has an error. Name the error (syntax/logical/runtime) and correct the code so that it compiles and runs correctly.

```
String s = "5";

double num = Integer.parseInt(s);

int cube = Math.pow(num, 3);

System.out.println(cube);
```

Ans : COMP 2011

Logical Error

```
String s = "5";

int num = Integer.parseInt(s);   //
Correct data type for num

double cube = Math.pow(num, 3);   //
Math.pow returns a double

System.out.println(cube);
```

Output: 125.0

50. A student is trying to convert a string present in numStr to a float value so that they can calculate the square of the converted value. However, the code has an error. Name the error (syntax/logical/runtime) and correct the code so that it compiles and runs correctly.

```
String numStr = "8.5";

int square = Float.parseFloat(numStr)
* Float.parseFloat(numStr);

System.out.println(square);
```

Ans : MAIN 2012

Error : Logical Error

```
String numStr = "8.5";

float square = Float.parseFloat(numStr) Float.
parseFloat(numStr);

System.out.println(square);
```

Output : 72.25

51. A student is trying to convert a string present in value to an integer value so that they can find the factorial of the converted value. However, the code has an error. Name the error (syntax/logical/runtime) and correct the code so that it compiles and runs correctly.

```
String value = "5";

double num = Integer.parseInt(value);

int factorial = 1;
```

NODIA

```
            double PF = 8.33 / 100 *
ob.basic;
            double NetPay = ob.basic +
DA + HRA;
           double Gross_Pay = NetPay -
PF;
          System.out.println("Basic Pay
= ₹" + s);
          System.out.println("Gross Pay
= ₹" + Gross_Pay);
      }
}
```

Variables used :

basic: Data member for storing the basic salary of an employee.

b : Formal argument of the constructor to initialize basic.

s : Local variable to store basic, used as an actual argument of an object.

DA, HRA, PF : Dearness Allowance, House Rent Allowance, and Provident Fund respectively.

NetPay : Total salary excluding deductions.

Gross_Pay : Gross salary after deducting PF.

75. (i) Write a Java class Number, define a constructor for the class to initialize a variable count as 0.
(ii) Write the method main() for the above class to create its object.

Ans : SQP 2013

(i)

```
class Number {
    private int count;

    public Number() {
        count = 0;
    }
}
```

(ii)

```
   public static void main() {
     /* Creating object of class
     Number */
     Number num = new Number();
     }
     /* End of class Number */
```

76. Write a Java class named MyClass. The description of the class is as follows :

Members	Member Name	Description
Instance variable	age	To store age of a student.
Constructor		Initializes age to 14.
Main method		Increment the age by 1. Display result.

Ans : COMP 2014

```
class MyClass {
     int age; /* Declaring instance
variable age */

    MyClass() {
        age = 14; /* Initializes age
to 14 */
    }
    public static void main() {
        MyClass obj = new MyClass();
/* Creates object */
        obj.age++; /* Increments age
by 1 */
              System.out.println("The
changed age = " + obj.age);
    }
}
```

77. Define a Java class Salesman that initializes the commission of the salesman and the sale done by him to zero, using constructor. Define main method that computes his commission, if his sales is ₹ 15000 and the commission is 2%.

Ans : SQP 2015

```
class Salesman {
    double sales, commission;

    Salesman() {
          sales = 0; /* Initializes
sales as 0 */
       commission = 0; /* Initializes
commission as 0 */
    }

    Salesman(double s, double c) {
         sales = s; /* Initializing
sales and commission with passed
values */
        commission = c;
    }

    public static void main() {
```

Downloaded from www.pdf.tube

FIFTEEN MARKS QUESTIONS

13. Write a class named Student with marks as its data member to calculate the total and percentage of marks obtained in three subjects. Use meaningful variables.
Total Marks = Sum of marks of three subjects
Percentage = (Total Marks / 300) × 100

Ans : COMP 2011

```
import java.util.*;

class Student {
    double marks1, marks2, marks3;

      // Constructor to initialize
marks for three subjects
     Student(double m1, double m2,
double m3) {
        marks1 = m1;
        marks2 = m2;
        marks3 = m3;
    }

     public static void main(String
args[]) {
               Scanner input = new
Scanner(System.in);

           // Input marks for three
subjects
           System.out.println("Enter
marks for Subject 1:");
           double subject1 = input.
nextDouble();
           System.out.println("Enter
marks for Subject 2:");
           double subject2 = input.
nextDouble();
           System.out.println("Enter
marks for Subject 3:");
           double subject3 = input.
nextDouble();

       // Create an object of Student
class
             Student student = new
Student(subject1,          subject2,
subject3);

            // Calculate Total and
Percentage
        double TotalMarks = student.
marks1 + student.marks2 + student.
marks3;
         double Percentage = (Total
Marks / 300) * 100;

        // Display Results
           System.out.println("Marks
Obtained in Subject 1: " + subject1);
           System.out.println("Marks
Obtained in Subject 2: " + subject2);
           System.out.println("Marks
Obtained in Subject 3: " + subject3);
           System.out.println("Total
Marks = " + TotalMarks);
      System.out.println("Percentage
= " + Percentage + "%");
    }
}
```

14. Create a class named Employee with basic as its data member to calculate the gross pay of an employee based on the following allowances and deductions. Use meaningful variable names.
Dearness Allowance (DA) : 25% of Basic Pay
House Rent Allowance (HRA) : 15% of Basic Pay
Provident Fund (PF) : 8.33% of Basic Pay
Net Pay : Basic Pay + Dearness Allowance + House Rent Allowance
Gross Pay : Net Pay – Provident Fund

Ans : MAIN 2012

```
import java.util.*;
class Employee {
    double basic;

      // Constructor to initialize
basic salary
    Employee(double b) {
        basic = b;
    }

     public static void main(String
args[]) {
       System.out.println("Enter the
basic pay");
     Scanner buf = new Scanner(System.
in);
        int s = buf.nextInt();
       Employee ob = new Employee(s);
         double DA = ob.basic * 25.0
/ 100;
          double HRA = 15.0 / 100 *
ob.basic;
```

NODIA

```
product_price: Price of the product
pack_type: Pack type of the product
flavor: Juice flavour
```

79. Define a Java class Employee with the following members :

```
Data members:
code, name, basic, hra, da, PF
Default constructor: Initializes code, basic, hra, da, and PF to zero.
Parameterized constructor: Initializes basic, computes hra, da, and PF as per the following criteria:
hra = 10% of basic
da = 55% of basic
PF = ₹ 1000
Main method: Computes and displays the net salary
NetSalary = basic + da + hra – PF
```

Ans : COMP 2017

```
class Employee {
    int code;
    String name;
    double basic, da, hra, PF;

    Employee() {
            /* Initializing instance variables to their default values */
        code = 0;
        name = "";
        basic = 0;
        da = 0;
        hra = 0;
        PF = 0;
    }

   Employee(int c, String nam, double basSalary) {
        code = c;
        name = nam;
        basic = basSalary;
        da = 55.0 / 100.0 * basic;
        hra = 10.0 / 100.0 * basic;
        PF = 1000;
    }
}

public static void main() {
    int cd = 1;
    String n = "RADHESHYAM";
    double salary = 25000.85;
    Employee emp1 = new Employee(cd, n, salary);
    double netSalary;
    netSalary = emp1.basic + emp1.da + emp1.hra - emp1.PF;
       System.out.println("EMPLOYEES PAYSLIP:");
    System.out.println();
      System.out.println("CODE: " + cd);
    System.out.println("NAME: " + n);
    System.out.println();
     System.out.println("NET SALARY: ₹" + netSalary);
}
```

80. Define a Java class Color as described below:

Members	Member Name	Description
Instance variables	colTop, colBottom	To store the top and bottom colors of the dress.
Default Constructor		Initialize colTop as Pink and colBottom as Black.
Parameterized Constructor		Initialize colTop and colBottom by accepted values.
Main method	main	Display the default colors as well as assigned ones through objects.

Ans : MAIN 2018

```
class Color {
    String colTop, colBottom;

    Color() {
       /* Defining default Constructor */
        colTop = "Pink";
        colBottom = "Black";
    }

    Color(String t, String b) {
           /* Defining parameterized
```

Click Here to Buy book on Amazon

```
rating <= 5.0)
            System.out.println("Super
Hit");
    }

    public static void main(String[]
args) {
              movieMagic mov = new
movieMagic();
        mov.accept();
        mov.display();
    }
}
```

Variables used :

mov : Object of the class

year : Year of the release of movie

title : Title of the movie

rating : Movie rating ranging from 0 to 5

84. (i) Write a Java class Days. Define its constructor that takes two integer values and stores in public variables day1 and day2.

(ii) Write another class Difference that creates an object of class Days, computes, and prints the difference in the two days day1 and day2.

Ans : SQP 2022

(i)

```
class Days {
   public int day1, day2; /* Declaring
instance variables */

    Days(int d1, int d2) {
       /* Initializing day1 by value
of d1 */
        day1 = d1;
       /* Initializing day2 by value
of d2 */
        day2 = d2;
    }
}
```

(ii)

```
class Difference {
    public static void main() {
         int dd1 = 15, dd2 = 25; /*
Creating object */
       Days obj = new Days(dd1, dd2);

          /* Computing difference in
number of days */
          int diff = obj.day2 - obj.
day1;
        System.out.println("Difference
in number of days = " + diff);
    }
}
```

Variables used :

1. day1, day2 : Public integer member variables of class Days.
2. d1, d2 : Formal parameters of the Days constructor.
3. dd1, dd2 : Actual parameters that store the number of days (first value) and number of days (second value).
4. diff : Stores the subtracted result of two days.

85. Design a class to overload a function polygon area() as follows:

(i) double area(double a, double b, double c) with three double arguments, returns the area of a scalene triangle using the formula :

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{where } s = \frac{a+b+c}{2}$$

(ii) Double area(int a, int b, int height) with three integer arguments, returns the area of a trapezium using the formula :

(iii) Double area(double diagonal1, double diagonal2) with two double arguments, returns the area of a rhombus using the formula :

$$\text{area} = \frac{1}{2} \times (\text{diagonal}\,1 \times \text{diagonal}\,2)$$

Ans : COMP 2023

The class is as follows :

```
public class Overload {
    double area(double a, double b,
double c) {
        double s = (a + b + c) / 2.0;
        double areaOfTriangle = Math.
sqrt(s * (s - a) * (s - b) * (s - c));
        return areaOfTriangle;
    }

     double area(int a, int b, int
height) {
        double ar = 1.0 / 2 * height
* (a + b);
        return ar;
    }

     double area(double diagonal1,
double diagonal2) {
           double ar = 1.0 / 2 *
```

```
        return Math.sqrt(number);
    }

    double calculateReciprocal() {
        return 1.0 / number;
    }

    public static void main(String[]
args) {
                Scanner input = new
Scanner(System.in);

         System.out.print("Enter an
integer number: ");
        int num = input.nextInt();

         NumberOperations obj = new
NumberOperations(num);

    double cube = obj.calculateCube();
          double squareRoot = obj.
calculateSquareRoot();
          double reciprocal = obj.
calculateReciprocal();

         System.out.println("Number:
" + num);
        System.out.println("Cube: "
+ cube);
         System.out.println("Square
Root: " + squareRoot);
     System.out.println("Reciprocal:
" + reciprocal);
    }
}
```

83. Define a class named movie-magic with the following specifications :
Instance variables/data members :
int year: To store the year of release of a movie.
String title: To store the title of the movie.
float rating: To store the popularity rating of the movie (minimum rating = 0.0 and maximum rating = 5.0).
Member methods :

(i) movieMagic() : Default constructor to initialize numeric data members to 0 and String data member to "".

(ii) void accept() : To input and store year, title, and rating.

(iii) void display() : To display the title of a movie and a message based on the rating as per the table below.

Rating Table :

Rating	Message to be displayed
0.0 to 2.0	Flop
2.1 to 3.4	Semi-hit
3.5 to 4.5	Hit
4.6 to 5.0	Super Hit

Write a main method to create an object of the class and call the above member methods.

Ans : MAIN 2021

The program code is as follows :

```
import java.util.Scanner;
public class movieMagic {
    int year;
    String title;
    float rating;

    movieMagic() {
        year = 0;
        title = "";
        rating = 0;
    }

    void accept() {
    Scanner scan = new Scanner(System.
in);
          System.out.println("Please
enter the year of release of the
movie");
        year = scan.nextInt();
          System.out.println("Please
enter the title of the movie");
        title = scan.next();
          System.out.println("Please
enter the rating of the movie");
        rating = scan.nextFloat();
    }
    void display() {
        System.out.println("TITLE: "
+ title);
         if (rating >= 0.0 && rating
<= 2.0)
          System.out.println("Flop");
           else if (rating >= 2.1 &&
rating <= 3.4)
            System.out.println("Semi-
hit");
          else if (rating >= 3.5 &&
rating <= 4.5)
          System.out.println("Hit");
          else if (rating >= 4.6 &&
```

NODIA

```
constructor */
        colTop = t;
        colBottom = b;
    }

    public static void main() {
            /* Creating object with
default colors */
        Color Dress1 = new Color();
      System.out.println("The colors
for Dress are " + Dress1.colTop + "
& " + Dress1.colBottom);

          String s1 = "WHITE", s2 =
"BLACK";

         /* Creating object with new
values */
        Color Dress2 = new Color(s1,
s2);
       System.out.println("The color
of second Dress is " + Dress2.colTop
+ " and " + Dress2.colBottom);
    }
}
```

Variables used :

1. colTop, colBottom : Instance variables to store colors of top and bottom of an object.
2. t, B : Formal parameters of the parameterized constructor to store passed values of colors of an object.
3. Dress1 : Dress object with default colors Pink and Black.
4. Dress2 : Dress object with colors White and Black.
5. s1, s2 : Store the new colors of Dress2.

81. Write a Java program to calculate and display the double, square, and half of an integer number using a constructor that takes the number as a parameter. The program should also print the computed results.

Ans : SQP 2019

```
class ComputeNumber {
    private int number; /* Declaring
number as integer member variable */
    public int d, s;
    public double h;

    ComputeNumber(int a) {
           /* Defining parameterized
constructor */
        number = a;
        d = 2 * number;
          h = 1 / 2.0 * number; /*
Computing half of the number */
           s = number * number; /*
Computing the square */
    }

    public static void main() {
       int n = 86; /* Creating object
*/
            ComputeNumber num = new
ComputeNumber(n);
       System.out.println("Double of
" + n + " = " + num.d);
       System.out.println("Square of
" + n + " = " + num.s);
         System.out.println("Half of
" + n + " = " + num.h);
    }
}
```

Variables used :

1. number : Private integer member variable on which calculations are to be performed.
2. d, s, h: Public member variables to store the double, square, and half of the number, respectively.
3. a : Formal parameter of constructor.
4. n : Actual parameter whose value (86) is copied to the formal parameter.
5. num : Object of class ComputeNumber.

82. Write a Java program to compute and display the cube, square root, and reciprocal of a given integer using a constructor that accepts the number as a parameter. The program should also print the computed results.

Ans : COMP 2020

```
import java.util.Scanner;

class NumberOperations {
    int number;

    NumberOperations(int num) {
        number = num;
    }

    double calculateCube() {
        return Math.pow(number, 3);
    }

    double calculateSquareRoot() {
```

Downloaded from www.pdf.tube

```
(diagonal1 * diagonal2);
        return ar;
    }
}
```

Variables used :

a, b, c : Length of sides of a triangle.

a, b, height : Length of two sides and the height of a trapezium.

diagonal1, diagonal2 : Diagonals of a rhombus.

86. Design a Java class named Box with the following specifications:

(i) The default constructor initializes the dimensions length, width, and height to zero.

(ii) A parameterized constructor accepts three double values representing the dimensions of the box.

Write a main method to create a Box object with dimensions 3.89 cm, 2.1 cm, and 1.5 cm, and calculate its volume.

Ans : SQP 2024

```
class Box {
    double length, width, height;

    Box() {
         length = 0; /* Initializing
member variables to zero */
        width = 0;
        height = 0;
    }

     Box(double l, double w, double
h) {
            /* Defining parameterized
constructor */
        length = l;
        width = w;
        height = h;
    }
public static void main() {
   Box obj = new Box(3.89, 2.1, 1.5);
    double volume;
     volume = obj.length * obj.width
* obj.height; /* Computing volume */
   System.out.println("Volume of the
box object = " + volume);
}
```

87. Consider a right-angled triangle whose longest side is given as 6 cm and the base as 4 cm. Write a Java program that initializes the given sides of the triangle in a constructor and computes and displays the third side.

Ans : MAIN 2018

```
class TriangleSide {
    double longestSide, base;

    TriangleSide(double l, double b)
{
        longestSide = l;
        base = b;
    }

    public static void main() {
        double s1 = 6, s2 = 4;
         TriangleSide triangle = new
TriangleSide(s1, s2);
        double t1, t2, s3;
         t1 = triangle.longestSide *
triangle.longestSide;
       t2 = triangle.base * triangle.
base;
         s3 = Math.sqrt(t1 - t2);
        System.out.println("The third
side = " + s3);
    }
}
```

Variables used :

longestSide, base: Member variables.

l, b : Formal parameters to store the passed values of sides of the triangle.

s1, s2 : Actual parameters that store the length of the longest side and the base of the triangle.

t1, t2 : Store intermediate results of calculating the third side.

s3 : Stores the computed third side by applying the formula:

$$(\text{hypotenuse})^2 = (\text{base})^2 + (\text{third side})^2$$

Therefore,

$$(\text{thirdside})^2 = (\text{hypotenuse})^2 - (\text{base})^2$$

Where hypotenuse is the longest side of a right-angled triangle.

88. Write a Java program to represent an isosceles triangle where the two equal sides are 5 cm each, and the base is 6 cm. Initialize the sides of the triangle using a constructor, and compute and display the perimeter of the triangle.

Ans : SQP 2015

```
class IsoscelesTriangle {
    double side1, side2, base;
       IsoscelesTriangle(double  s1,
double s2, double b) {
        side1 = s1;
```

NODIA

17. JAVA STATEMENT TO READ A COMPLETE STRING FROM KEYBOARD

```
String name = var.readLine();
```

18. JAVA STATEMENT TO CONVERT THE STRING OF CHARACTERS TO INTEGER TYPE

```
String number = var.readLine();
int num = Integer.parseInt(number);
```

19. JAVA STATEMENT TO CONVERT THE STRING OF CHARACTERS TO FLOATING TYPE

```
float num = Float.parseFloat(number);
```

20. JAVA STATEMENT TO CONVERT THE STRING OF CHARACTERS TO DOUBLE TYPE

```
double num = Double.parseDouble(number);
```

21. JAVA STATEMENT TO CONVERT THE STRING OF CHARACTERS TO LONG TYPE

```
long num = Long.parseLong(number);
```

22. CONVERTING NUMERIC STRINGS TO PRIMITIVE NUMBER DATATYPE USING PARSING METHODS

(i) Convert string to integer type :
```
int x = Integer.parseInt(str);
```
(ii) Convert string to long type:
```
long d = Long.parseLong(str);
```
(iii) Convert string to float type:
```
float y = Float.parseFloat(str);
```
(iv) Convert string to double type:
```
double z = Double.parseDouble(str);
```
In the above statements, str is a string variable that is either initialized as a numeric value or gets a numeric value inputted from the user.
e.g. `String str = "123";`

23. CONVERTING STRING OBJECTS TO NUMERIC OBJECTS USING VALUEOF() METHOD

(i) Convert string to an integer :
```
int x = Integer.valueOf(str);
```
(ii) Convert string to a long type :
```
long l = Long.valueOf(str);
```
(iii) Convert string to float type:
```
float v = Float.valueOf(str);
```
(iv) Convert string to double type :
```
double d = Double.valueOf(str);
```

24. PRE-DEFINED FUNCTIONS FOR WRAPPER CLASS CHARACTER

(i) Boolean isDigit(char ch)

This function returns true if the character is a digit; else it returns false.

e.g.,

```
boolean v = Character.isDigit('x');
```

For the above example, v stores false.

(ii) Oolean isLetter(char ch)

This method returns true if ch is a letter; else it returns false.

e.g.,

```
if (Character.isLetter('1'))
        System.out.print("It's a letter");
else
        System.out.print("It's a digit");
```

Output: It's a digit.

(iii) Boolean isLetterOrDigit(char ch)

This method returns true if ch is either a letter or a digit; otherwise, it returns false.

e.g.,

```
if (Character.isLetterOrDigit('Q'))
    System.out.print("It is either alphabet or a digit");
else
        System.out.print("It is a special character");
```

Output: It is either alphabet or a digit.

(iv) Boolean isLowerCase(char ch)

This method returns true if ch is a lowercase alphabet, else it returns false.

```
char y = 'a';
boolean d = Character.isLowerCase(y);
System.out.print(d);
```

Output : true

(v) Boolean isUpperCase(char ch) : This method returns true if the character ch is an uppercase letter, else it returns false.

e.g.,

```
java
if (Character.isUpperCase('D'))
 System.out.print("It's an UpperCase letter");
```

Click Here to Buy book on Amazon

```
      // Constructor to initialize
numbers
   public Numbers(double num1, double
num2, double num3) {
        this.number1 = num1;
        this.number2 = num2;
        this.number3 = num3;
         this.total = num1 + num2 +
num3;
        this.average = total / 3;
    }

    // Method to display results
    public void displayResults() {
       System.out.println("Number 1:
" + number1);
       System.out.println("Number 2:
" + number2);
       System.out.println("Number 3:
" + number3);
        System.out.println("Total: "
+ total);
        System.out.println("Average:
" + average);
    }

    public static void main(String[]
args) {
     Numbers numbers = new Numbers(10,
20, 30); // Example input
        numbers.displayResults();
    }
}
```

92. Write a class named Box with length, width, and height as its data members. Calculate the volume and surface area of the box.

Formulas : Volume = length×width× height
Surface Area = 2 × (length × width + width × height + height × length)

Ans : MAIN 2021

```
class Box {
     private double length, width,
height, volume, surfaceArea;

      // Constructor to initialize
dimensions
    public Box(double length, double
width, double height) {
        this.length = length;
        this.width = width;
        this.height = height;
        this.volume = length * width
* height;
       this.surfaceArea = 2 * (length
* width + width * height + height *
length);
    }

    // Method to display results
    public void displayResults() {
         System.out.println("Length:
" + length);
        System.out.println("Width: "
+ width);
         System.out.println("Height:
" + height);
         System.out.println("Volume:
" + volume);
         System.out.println("Surface
Area: " + surfaceArea);
    }

     public static void main(String[
] args) {
         Box box = new Box(5, 4, 3);
// Example input
        box.displayResults();
    }
}
```

NODIA

www.nodia.in

```
        side2 = s2;
        base = b;
    }

    double calculatePerimeter() {
        return side1 + side2 + base;
    }

    public static void main(String[] args) {
        IsoscelesTriangle triangle = new IsoscelesTriangle(5.0, 5.0, 6.0);
        double perimeter = triangle.calculatePerimeter();
        System.out.println("The perimeter of the isosceles triangle is: " + perimeter + " cm");
    }
}
```

89. Write a Java program to compute and display the area and perimeter of a rectangle using a constructor that accepts its length and width as parameters.

Ans : COMP 2012

```
class Rectangle {
    private double length, width, area, perimeter;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
        this.area = length * width;
        this.perimeter = 2 * (length + width);
    }

    public void display() {
        System.out.println("Length: " + length);
        System.out.println("Width: " + width);
        System.out.println("Area: " + area);
        System.out.println("Perimeter: " + perimeter);
    }

    public static void main(String[] args) {
        Rectangle rect = new Rectangle(5, 3); // Example input
        rect.display();
    }
}
```

90. Write a Java program to compute and display the simple interest using a constructor that accepts the principal, rate of interest, and time as parameters.

Ans : MAIN 2014

```
class SimpleInterest {
    private double principal, rate, time, interest;

    public SimpleInterest(double principal, double rate, double time) {
        this.principal = principal;
        this.rate = rate;
        this.time = time;
        this.interest = (principal * rate * time) / 100;
    }

    public void display() {
        System.out.println("Principal: " + principal);
        System.out.println("Rate of Interest: " + rate + "%");
        System.out.println("Time: " + time + " years");
        System.out.println("Simple Interest: " + interest);
    }

    public static void main(String[] args) {
        SimpleInterest si = new SimpleInterest(5000, 5, 2); // Example input
        si.display();
    }
}
```

91. Write a class named Numbers with three numbers as its data members. Calculate the total and average of the numbers.

Ans : COMP 2015

```
class Numbers {
    private double number1, number2, number3, total, average;
```

Downloaded from www.pdf.tube

7. COMPOSITE DATA TYPES

The structure of such data types makes use of primitive types. It is designed by the programmer and thus it is called user-defined data types.

One or more than one similar or different primitive types makes it a composite data type (also called Reference data types).

The reference types are also known as composite data types.

(Refer Chapter-2 also)

Example of composite data type (class):

```
class ComplexNumber {
    int real_part, imag_part;
}
```

8. CLASS AS A COMPOSITE (USER-DEFINED) DATA TYPE

A class is a construct that binds one or more primitive types together to be used as a single data type. These primitive types integrated in the class can be of

(i) Primitive type variables (called member variables)

(ii) Primitive type methods (called member methods)

9. STORAGE SPACE OCCUPIED BY A CLASS

A class occupies the sum total space of all the primitive types it is composed of, because each type occupies separate space from the pool of free memory.

10. WRAPPER CLASSES

Wrapper classes in Java are used to wrap primitive data types into objects. These classes provide a way to use primitive data types as objects, enabling operations like collection storage, method calls, and more, which are not directly possible with primitives.

10.1 Purpose of Wrapper Classes

(i) To treat primitive types as objects.

(ii) Useful in data structures like ArrayList, HashMap, etc., which work with objects.

(iii) Provide utility methods for converting between types, parsing strings, and performing operations on wrapped values.

11. LIST OF WRAPPER CLASSES AND THE PRIMITIVE TYPES THAT THEY CONTAIN

Wrapper classes encapsulate primitive data types, allowing them to be treated as objects in Java. Below is the list of wrapper classes and their associated primitive types:

Simple type	Wrapper Class
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

12. STREAM

A stream in Java represents a flow of data in the form of bytes, allowing data to be transferred between input/output devices and the program's main memory. Streams are broadly categorized into Input Streams and Output Streams.

12.1 Input Stream

This stream is ready to supply input data. It corresponds to keyboard input or other input sources.

12.2 Output Stream

Output stream is ready to accept output data. print(), println() methods are in the class printStream.

System.out is an object of the printStream class.

13. BUFFERED READER

A class that supports a buffered InputStream.

14. INPUTSTREAMREADER

A sub-class that converts bytes to characters.

15. JAVA STATEMENT TO CREATE A VARIABLE OF BUFFEREDREADER CLASS

```
BufferedReader var = new BufferedReader(new InputStreamReader(System.in));
```

Note: var is a user-defined variable name linked to the keyboard through System.in.

16. JAVA STATEMENT TO READ CHARACTER FROM KEYBOARD

```
char ch = (char) var.read();
```

NODIA

CHAPTER 5

LIBRARY CLASSES

SUMMARY

1. LIBRARY CLASSES OF JAVA

Library classes in Java are pre-defined classes that are included in an application program by default. These are also known as standard Java classes or built-in Java classes. They contain member variables, constructors, and methods like ordinary classes created by programmers. These classes are organized into packages.

2. PACKAGES

Packages group classes together for better organization and re-usability. Classes in a package can be used in other programs without physically copying them into the program.

2.1 Importing Packages

All standard library classes of Java are stored in a named package. To make certain classes or all classes of a package visible (accessible) in another class, an import statement is written to indicate the Java compiler to import specified classes of the package.
Example : import java.util.

3. COMMONLY USED JAVA PACKAGES

3.1 java.lang

(i) All basic functions of the language are stored in this package.

(ii) It can be imported by writing: import java. lang.

(iii) This package is imported in Java by default.

Key Classes in java.lang

(i) Wrapper classes : A collection of classes used to wrap variables of simple data types. Boolean, Character, Double, Float, Integer, and Long are wrapper classes.

(ii) Strings : String class and StringBuffer class.

(iii) Math : A library of math routines and values as pi.

(iv) System : Contains three pre-defined variables: in, out, and err.

3.2 java.io

This package provides a set of InputStreams and OutputStreams used to read and write data to a file or input and output devices.

4. CREATING PACKAGES

To create a package, use the package keyword followed by the package name at the beginning of the file.

```
package NameOfPackage;
class ClassName {
     // Code
}
```

4.1 Importing Class from User-defined Package

import name of Package.className;

5. DATA TYPE

The type of data values required to be operated upon are categorized as different data types. The variables in which these values get stored are specified with their data types.
For example, 10 is an integer that gets stored in a variable a.
It is declared as :

```
java
int a = 10;
```

6. PRIMITIVE DATA TYPES

These are the inbuilt data types which serve as the basic types for the derived (reference) types.
Examples of primitive type :
byte, short, int, long, char, boolean, float, and double are primitive types.

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

(vi) Boolean isWhitespace(char ch)

This method returns true if ch is a space.

e.g.,

```
char x = ' ';
boolean y = Character.
                    isWhitespace(x);
```

Here, y stores true.

(vii) char toLowerCase(char ch)

This method converts the letter ch to a corresponding lowercase letter.

e.g.,

```
java
char x = Character.toLowerCase('A');
System.out.print(x);
```

Output: a

(viii) char toUpperCase(char ch)

This method converts the letter ch to uppercase.

e.g.,

```
char ch = 'a';
if (Character.isLowerCase(ch)) {
ch = Character.toUpperCase(ch);
System.out.print(ch);
}
```

Output: A

25. AUTO BOXING

Process of converting primitive datatype value to corresponding wrapper class is called autoboxing.

Example of auto-boxing

```
// Autoboxing of char value
Character y = 'a';
// Unboxing character
char c = y;
```

26. UNBOXING

Process of converting object of a wrapper class to its corresponding primitive data type is known as unboxing.

Example of unboxing:

```
// Creating an Integer object
Integer i = new Integer(20);
// Unboxing the object
int x = i;
```

Download INDIA's Best Study app NODIA

MULTIPLE CHOICE QUESTION

1. What is the output of the below code segment?
String x = "I love programming"?
StringBuffer y = new StringBuffer(x);
StringBuffer z = y.reverse();
System.out.print(z);

(a) programming love I

(b) evol I programming

(c) gnimmargorp love I

(d) gnimmargorp evol I

Ans : MAIN 2024

(d) gnimmargorp evol I

The given code creates a StringBuffer object y initialized with the string "I love programming". The reverse() method of StringBuffer reverses the order of characters in the string, resulting in "gnimmargorp evol I". Finally, the reversed string is printed using System.out.print(z). Hence, the output is option (d).

2. Which of the following is the wrapper class for simple data type-char?

(a) String

(b) Char

(c) Character

(d) Float

Ans : MAIN 2024

(c) Character

In Java, the wrapper class for the primitive data type char is Character. Wrapper classes convert primitive types into objects, allowing char values to be used in object-oriented features like collections and generics. String is not a wrapper class but represents a sequence of characters, and Float is the wrapper for float. Char is invalid in Java.

3. Which of the following is a super class of all other Java classes?

(a) Object

(b) Math

(c) System

(d) Import

Ans : SQP 2015

(a) object

In Java, Object is the superclass of all other classes. It is the root of the class hierarchy,

Downloaded from www.pdf.tube

21. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to print whether a number x is odd or even.

```
void checkOddEven(int x)
{
            System.out.println("Odd");
// (1)
    if (x % 2 == 0)
          System.out.println("Even");
// (2)
    else
            System.out.println("Odd");
// (3)
}
```

(a) (2) (3) (1)
(b) (1) (2) (3)
(c) (3) (2) (1)
(d) (2) (1) (3)

Ans : SQP 2013

(b) (1) (2) (3)
To correctly determine and print whether a number x is odd or even, the correct order of statements should be:

(i) (2) Check if x % 2 == 0 and print "Even" if the condition is true.

(ii) (3) If the condition is false (in the else block), print "Odd".

(iii) (1) The standalone "Odd" statement is unnecessary here and could be omitted for clarity. If it must remain, it would cause confusion, as it prints "Odd" regardless of the actual condition.

However, logically, the core part of the function involves just the conditional check (statements 2 and 3). Statement 1 doesn't contribute meaningfully in this context.

22. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to calculate the sum of all elements in an array arr.

```
void calculateSum(int[] arr)
{
     for (int i = 0; i < arr.length;
i++)
         sum += arr[i];              //
(1)
    System.out.println(sum);         //
(2)
    int sum = 0;                     //
(3)
}
```

Options:

(a) (3) (1) (2)
(b) (1) (3) (2)
(c) (2) (3) (1)
(d) (3) (2) (1)

Ans : COMP 2014

(a) (3) (1) (2)
To correctly calculate the sum of all elements in an array and print the result, the statements should be ordered as follows:

(i) (3) Initialize the sum variable with int sum = 0; to store the cumulative total.

(ii) (1) Use a loop to iterate through each element of the array arr and add its value to sum:

```
java
    for (int i = 0; i < arr.length;
i++)
        sum += arr[i];
```

(iii) (2) Print the total sum after the loop finishes using System.out.println(sum);.

This order ensures the sum variable is initialized before use and that all elements are added to it before displaying the result.

23. **Assertion (A) :** Wrapper classes in Java are used to treat primitive types as objects.
Reason (R) : Wrapper classes enable the use of primitive data types in data structures like Array List and Hash-map.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : MAIN 2015

(a) Both A and R are true, and R is the correct explanation of A.
Wrapper classes in Java (e.g., Integer, Double, Character) allow primitive types to be treated as objects. This is essential because data structures like ArrayList and HashMap can only store objects, not primitives. Hence, the assertion (A) is true. The reason (R) accurately explains the need for wrapper classes, making both the assertion and the reason true, with R correctly explaining A.

NODIA

(c) 1
The Character.toLowerCase() method converts an uppercase letter to its lowercase equivalent. However, if the character is not an uppercase letter (e.g., a digit like '1'), the method returns the character unchanged. Since '1' is a digit, the output will remain '1'. There is no error in the code, and the output is not a letter or boolean value.

10. What will be the output for the following ode?
System.out.print(Character.toLowerCase('A') + 1);
(a) 66
(b) B
(c) Error
(d) 98

Ans : COMP 2002

(d) 98
The Character.toLowerCase('A') method converts the uppercase letter 'A' to its lowercase equivalent 'a'. The ASCII value of 'a' is 97. When you add 1, it becomes 98. However, since the addition involves an implicit conversion to an int, the output will be the integer 98..

11. Return data type of isLetter(char) is _____.
(a) Boolean
(b) Boolean
(c) Bool
(d) Char

Ans : MAIN 2003

(b) Boolean
The return data type of the Character. isLetter(char) method is boolean (lowercase), which is a primitive data type in Java. It checks whether the provided character is a letter (a-z, A-Z) and returns true or false accordingly. Boolean (uppercase) is the wrapper class, bool is invalid in Java, and char is unrelated.

12. Corresponding wrapper class of float data type is _____.
(a) FLOAT
(b) Float
(c) Float
(d) Floating

Ans : SQP 2004

(c) Float
In Java, the wrapper class for the primitive data type float is Float. Wrapper classes are used to convert primitive data types into objects. This enables the use of float values in contexts requiring objects, such as collections or generics. FLOAT and Floating are invalid, while float refers to the primitive type itself.

13. A student wants to check if a character c is a digit. He writes the following incorrect statement:
`boolean result = c.isDigit();`
What will be the correct statement?
A. Boolean result = Character.isDigit(c);
B. Boolean result = c.isDigit();
C. Boolean result = isDigit(c);
D. Both A and C
(a) Only A
(b) Only B
(c) Both A and C
(d) All three

Ans : COMP 2005

(a) Only A
The correct way to check if a character c is a digit is by using the static method Character. isDigit(c). This method belongs to the Character class, not the char primitive type. Option B is invalid because char does not have an isDigit() method, and option C is invalid because isDigit(c) without Character is not defined in Java.

14. A programmer wants to convert a numeric string numStr to an integer. He writes the following incorrect statement:
`int num = Integer.convert(numStr);`
What will be the correct statement?
A. int num = Integer.parseInt(numStr);
B. int num = Integer.valueOf(numStr);
C. int num = Integer.convert(numStr);
D. Both A and B
(a) Only A
(b) Only B
(c) Both A and B
(d) All three

Ans : MAIN 2006

(c) Both A and B
The correct ways to convert a numeric string numStr to an integer in Java are:
Using Integer.parseInt(numStr), which converts the string directly to an int.
Using Integer.valueOf(numStr), which returns an Integer object that can be auto-unboxed to an int.

Click Here to Buy book on Amazon

meaning every class in Java directly or indirectly inherits from Object. This provides common methods like toString(), equals(), and hashCode() to all classes. Math, System, and Import are unrelated to class hierarchy and do not serve as superclasses.

4. Which of the following is the output of the below code snippet?
double x = -3.142;
int y = (int) Math.abs(x) + (int) x;
System.out.print(y);
(a) 0
(b) 6.284
(c) 6.0
(d) -6

Ans : COMP 2022

(a) 0
The code calculates the absolute value of x using Math.abs(x), which returns 3.142. This is then cast to an int, resulting in 3. The value of x is -3.142, and casting it to an int results in -3. Adding these gives (3 + (-3) = 0). Hence, the output is 0.

5. Which of the following methods of Java returns the nearest and larger whole number which is greater than or equal to the argument?
(a) Ceil
(b) Floor
(c) Trim
(d) Max

Ans : MAIN 2023

(a) Ceil
In Java, the Math.ceil() method returns the smallest integer greater than or equal to the given argument. It rounds up to the nearest whole number. For example, Math.ceil(3.2) returns 4. Math.floor() rounds down, trim() is a String method to remove whitespace, and max() returns the larger of two numbers.

6. When an object of a class is converted to a simple data type, it is called _____
(a) Unboxing
(b) Boxing
(c) Explicit type conversion
(d) Implicit type conversion

Ans : SQP 2021

(a) Unboxing
Unboxing in Java is the process of converting an object of a wrapper class to its corresponding primitive data type. For example, converting a Integer object to an int value is unboxing. This is the reverse of boxing, where a primitive type is converted into its wrapper object. Explicit and implicit type conversions refer to different concepts.

7. Which of the following would be the result if you execute the following statement?
System.out.print(Character.isDigit(-1));
(a) False
(b) True
(c) Error
(d) None of the above

Ans : COMP 2020

(c) error
The Character.isDigit() method checks if a given character (as an integer value) is a digit (0-9). The value -1 is not a valid Unicode code point for a digit, so the method returns false. There is no error in execution, and the output is not true since -1 is not a digit..

8. Which of the following will be the output if you execute the following code segment?
System.out.print(Character.isLetterOrDigit('1'));
(a) False
(b) True
(c) Error
(d) None

Ans : MAIN 2000

(b) true
The Character.isLetterOrDigit() method checks whether the given character is a letter (a-z, A-Z) or a digit (0-9). The character '1' is a digit, so the method returns true. There is no error in the code, and the output is not false because '1' is a valid digit.

9. What will be the output for the following code?
System.out.print(Character.toLowerCase('1'));
(a) Error
(b) A
(c) 1
(d) true

Ans : SQP 2001

NODIA

18. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to calculate the area of a rectangle with length l and breadth b.

```
void calculateArea(int l, int b)
{
     System.out.println(area);     //
(1)
    int area = l * b;           // (2)
            System.out.println("Area
calculated!");  // (3)
}
```

(a) (2) (1) (3)
(b) (3) (2) (1)
(c) (1) (2) (3)
(d) (2) (3) (1)

Ans : SQP 2010

(d) (2) (3) (1)
To calculate and display the area of a rectangle with length l and breadth b, the steps should follow this order:

(i) First, calculate the area using int area = l * b;.
(ii) Print the message "Area calculated!" to confirm that the area is computed.
(iii) Finally, display the calculated area using System.out.println(area);.

This ensures that the area is computed before attempting to print it, avoiding any runtime errors..

19. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to reverse a string s.

```
void reverseString(String s)
{
       System.out.println(reversed);
// (1)
    String reversed = "";          //
(2)
     for (int i = s.length() - 1; i
>= 0; i--)
            reversed += s.charAt(i);
// (3)
}
```

(a) (2) (3) (1) (b) (1) (2) (3)
(c) (3) (2) (1) (d) (2) (1) (3)

Ans : COMP 2011

(a) (2) (3) (1)
To reverse a string s and display the reversed result, the statements should be ordered as follows:

(i) (2) Initialize the reversed string with an empty value: String reversed = "";.
(ii) (3) Use a loop to iterate through the string s in reverse order, appending each character to the reversed string:

```
for (int i = s.length() - 1; i >= 0;
i--)
     reversed += s.charAt(i);
```

(iii) (1) Print the reversed string using System.out.println(reversed);.

This order ensures that the reversed string is initialized before being used and that it is fully computed before printing.

20. Consider the following program segment in which the statements are jumbled. Choose the correct order of statements to find the factorial of a number n.

```
void factorial(int n)
{
   System.out.println(fact);       //
(1)
    for (int i = 1; i <= n; i++)
       fact *= i;                  //
(2)
   int fact = 1;                   //
(3)
}
```

(a) (3) (2) (1)
(b) (1) (2) (3)
(c) (2) (3) (1)
(d) (3) (1) (2)

Ans : MAIN 2012

(a) (3) (2) (1)
To calculate the factorial of a number n and display the result, the statements should be ordered as follows:

(i) (3) Initialize the fact variable to 1 with int fact = 1;. This sets the initial value for multiplication.
(ii) (2) Use a loop to calculate the factorial by multiplying fact with each integer from 1 to n:

```
for (int i = 1; i <= n; i++)
   fact *= i;
```

(iii) (1) Print the calculated factorial using System.out.println(fact);.

This order ensures that the variable fact is initialized before use, and the factorial is fully computed before it is printed.

Downloaded from www.pdf.tube

(iv) char chr = Character.toLowerCase(ch);
(v) char chr = Character.toUpperCase(ch);

48. (i) Write Java statement to read (input) a character from the keyboard.
(ii) Write Java statement to assign characters a, e, i, o, u to variable chr.

Ans : SQP 2007

(i)

```
char ch;
BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));
ch = (char) buf.read();
```

(ii)

```
char chr[] = {'a', 'e', 'i', 'o', 'u'};
```

49. Write Java statements to perform the following tasks of converting strings to numbers or vice-versa:
(i) Convert a string number stored in x variable to an integer.
(ii) Convert a string number stored in y variable to long type.
(iii) Convert a string number stored in z variable to double type.
(iv) Convert string number stored in u variable to float type.

Ans : MAIN 2008

(i) int n = Integer.parseInt(x);
(ii) long n = Long.parseLong(y);
(iii) double n = Double.parseDouble(z);
(iv) float n = Float.parseFloat(u);

50. Why is a class known as a composite data type?

Ans : SQP 2005

A class is called a composite data type because it binds up (encapsulates) one or more primitive types together as a single data type.
e.g.,

```
java
  class Abc {
  int x;
  float y;
  }
```

51. (i) What is a wrapper class? Give an example.
(ii) What is the purpose of the new operator?

Ans : MAIN 2004

(i) A wrapper class is a collection of classes used to wrap variables of simple data type.
Example : Boolean, Character, Double, Float, Integer, and Long are wrapper classes.
(ii) new operator is a keyword and a Java operator that allocates memory for an object at runtime and returns a reference (address) of this memory.

52. How much storage space does a class type variable reserve for itself?

Ans : COMP 2003

A class type variable (which is an object) reserves the sum total of the storage space (in bits/bytes) occupied by each of the member variables and member functions.
For example:

```
class Abc {
    int a;
    float b;
    void function1() {
        a = 5; b = 10;
        System.out.println(a);
        System.out.print(b);
    }
}
```

An object of the above class Abc will occupy 4 bytes for its integer variable a and 4 bytes for its float variable b. In total, 8 bytes are reserved for the object variable of class Abc.

53. Write the differences between a primitive type and a composite data type.

Ans : SQP 2002

A primitive data type is an inbuilt simple data type of Java.
The number of bits occupied by each of such primitive data types is fixed and cannot be altered by the programmer.
A composite data type is defined by a programmer in the program. It comprises one or more primitive types.
The primitive data type and the number of variables of each type decide the number of bits/bytes reserved for a composite type (class).

54. Illustrate the above difference through examples.

Ans : COMP 2001

Example of primitive data type:

NODIA

24. **Assertion (A) :** The method Character.isDigit(char ch) is used to check if a character is a digit.
Reason (R) : The Character.isDigit() method returns false for alphabetic characters and true for numerical characters.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : SQP 2016

(a) Both A and R are true, and R is the correct explanation of A.
The Character.isDigit(char ch) method in Java checks whether the given character is a digit (0-9). It returns true for numerical characters and false for alphabetic or non-numerical characters. The reason (R) accurately explains the functionality of the method, making both the assertion (A) and reason (R) true, with R correctly explaining A.

25. **Assertion (A) :** The parseInt() method in the Integer class is used to convert a string to an integer.
Reason (R) : The parseInt() method throws a NumberFormatException if the string cannot be converted to a valid integer.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : COMP 2017

(a) Both A and R are true, and R is the correct explanation of A.
The parseInt() method in the Integer class is used to convert a string representation of a number into an integer. If the string contains non-numeric characters or is invalid for conversion, the method throws a NumberFormatException. Thus, the assertion (A) is true, and the reason (R) correctly explains the behavior of the parseInt() method, making both true with R being the correct explanation of A.

26. **Assertion (A) :** The method Character.toUpperCase(char ch) converts a character to its uppercase equivalent.
Reason (R) : The toUpperCase() method works only on lowercase characters and does not affect other characters.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : MAIN 2018

(c) A is true, but R is false.
The Character.toUpperCase(char ch) method in Java converts a character to its uppercase equivalent if it is a lowercase letter. If the character is already uppercase or not a letter, it returns the character unchanged. Thus, the assertion (A) is true. However, the reason (R) is false because toUpperCase() does not exclusively work on lowercase characters; it simply leaves non-lowercase characters unaffected.

27. **Assertion (A) :** Streams in Java are used to transfer data between input/output devices and the program's main memory.
Reason (R) : Input streams supply data, and output streams accept data for display or storage.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : COMP 2019

(a) Both A and R are true, and R is the correct explanation of A.
Streams in Java are used for transferring data between input/output devices and the program's main memory. Input streams are used to read data (supply data to the program), while output streams are used to write data (accept data for display or storage). The reason (R) accurately explains the assertion (A), making both true with R as the correct explanation of A.

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

TWO MARKS QUESTIONS

28. On what basis are the classes grouped?

Ans : SQP 2020

Classes are grouped based on their similarity in properties and behaviors. A class represents a blueprint for a group of objects that share common characteristics, such as attributes (data members) and behaviors (methods). This grouping helps in organizing code, achieving modularity, and reusing code efficiently in object-oriented programming. For example, a Vehicle class groups objects like Car, Bike, and Truck based on shared attributes like speed, color, and methods like move() or stop().

29. What are library classes? Give an example.

Ans : MAIN 2021

Library classes of Java: Pre-defined classes of Java which get included in an application program itself are called the library classes. They are also known as standard Java classes or built-in Java classes. These library classes have built-in member variables, constructors, and methods like ordinary classes developed by programmers.

30. Write a Java statement to input/read the following from the user using the keyboard:
(i) Character
(ii) String

Ans : COMP 2022

(i) char x = (char) buf.read();Where,buf is an object of BufferedReader class.
(ii) String y = buf.readLine();

31. State the method that:
(i) Converts a string to a primitive float data type.
(ii) Determines if the specified character is an uppercase character.

Ans : MAIN 2023

(i) parseFloat()
(ii) isUpperCase()

32. What is the data type that the following library functions return?
(i) isWhitespace(char ch)
(ii) Math.random()

Ans : COMP 2024

(i) Boolean
(ii) Double

33. Where are these classes grouped together and placed?

Ans : COMP 2024

These library classes are grouped together as packages. A collection of such classes in packages are also known to comprise the "class Library" of Java.

34. Write the benefit of grouping the classes as packages?

Ans : SQP 2020

Benefit of grouping the classes as packages:
(i) Efficient management of classes : As we know, Java has defined numerous classes. Each class containing numerous functions (methods), so, it becomes difficult to manage them. Classifying them into packages has brought about efficient handling of these classes.
(ii) Visibility Control : Classes can be defined in a package, and it can be checked that they will not be accessible outside that package. Even the accessibility of class members can be restricted.

35. What is the re usability feature of Java and how does Java implement it?

Ans : MAIN 2019

The term Re usability means that the already written code (program) can be reused by other programs.
In Java, it is implemented by using the concept of packages. The classes defined in packages can also be used by other programs without actually physically copying them.

36. What are the two categories of packages?

Ans : COMP 2018

API (Application Programming Interface) packages and User-defined packages are the two classifications of packages in Java.

37. What is meant by a package? Name any two Java Application Programming Interface packages.

Ans : SQP 2019

A package is a namespace (like a folder) that contains a set of related classes together.

NODIA

Two Java API packages are:

(i) java.io

(ii) java.math

38. What are Java API packages?

Ans : MAIN 2017

Java API has grouped a large number of classes into many packages according to the functions performed by them. The classes of these packages can be imported to our program according to need.

39. What do you understand by the term "User-defined packages"?

Ans : COMP 2016

When a programmer needs to group his own classes that he has designed for his applications and can create his own package using the keyword package followed by a package name. These are called User-defined packages.

40. How can you access the classes stored in an API package?

Ans : SQP 2015

Java statement for accessing classes in a package:
General Format:

```
import APIpackageName.classname;
import APIpackageName.*;
```

Note: If a class is contained in a package within another package, the package names are also separated by dot (.).

Example :

```
import java.io.*;
import java.util.Date;
```

41. Name the API package that contains wrapper classes.

Ans : MAIN 2014

java.lang

42. Write the return data type of the following library functions:

(i) CompareTo()

(ii) Equals()

Ans : COMP 2013

(i) The return data type of compareTo() is integer (int).

(ii) The return data type of equals() is boolean.

Download INDIA's Best Study app NODIA

43. Which package has defined the class System?

Ans : SQP 2012

System class is defined in the java.lang package.

44. What are the three objects of the System class? What are their types?

Ans : MAIN 2011

System.in : Here, in is of InputStream type class.
System.out and System.err: Here, out and err are of type printStream.

45. State the data type and value of y after the following is executed:

```
char x = '7';
y = Character.isLetter(x);
```

Ans : SQP 2010

Data type of y is boolean, and value of y = false.

46. Which package needs to be imported whenever you accept inputs from the keyboard using BufferedReader class?

Ans : COMP 2009

java.io package needs to be imported as follows:

```
java
import java.io.*;
```

47. Write Java statements to perform the following tasks on characters:

(i) Check if a character ch is an alphabet or not. Store the result in variable var.

(ii) Check if a character ch is a digit. Store the result in variable var.

(iii) Store the boolean result of checking a character as:
- (a) an uppercase letter
- (b) a lowercase letter
- (c) a space
- (d) an alphabet or digit.

(iv) Convert a character ch to lowercase.

(v) Convert a character ch to uppercase.

Ans : COMP 2006

(i) Boolean var = Character.isLetter(ch);

(ii) Boolean var = Character.isDigit(ch);

(iii) (a) Boolean var = Character.is Upper Case (ch);
- (b) Booleanresult = .is Lower Case (ch);
- (c) Booleanresult = Character.is Whitespace (ch);
- (d) Booleanb = Character.is Letter Or Digit (ch);

NODIA

Option C is invalid because the method Integer.convert() does not exist in Java.

15. A developer wants to read a string from the keyboard using BufferedReader. He writes the following incorrect statement:

```
BufferedReader br = new BufferedReader(System.in);
String input = br.read();
```

What will be the correct statement to read a string?

(a)

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String input = br.readLine();
```

(b)

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String input = br.read();

(c)

BufferedReader br = System.in;
String input = br.readLine();

(d) A and C

(a) Only A

(b) Only B

(c) Both A and C

(d) All three

Ans : SQP 2007

(a) Only A

The correct way to read a string from the keyboard using BufferedReader is:

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String input = br.readLine();
```

(i) BufferedReader requires a Reader object, and InputStreamReader(System.in) converts the input stream to a character stream.

(ii) The readLine() method reads a complete line of input as a string.

Option B is invalid because read() returns an integer (the ASCII value of a single character). Option C is invalid because System.in cannot directly be assigned to BufferedReader.

16. A programmer needs to check if a character ch is a whitespace character. He writes the following incorrect statement:

```
boolean isSpace = Character.isWhite(ch);
```

What will be the correct statement?

A. Boolean isSpace = Character.isWhitespace(ch);

B. Boolean isSpace = Character.isWhite(ch);

C. Boolean isSpace = ch.isWhitespace();

D. Both A and C

(a) Only A

(b) Only B

(c) Both A and C

(d) All three

Ans : COMP 2008

(a) Only A

The correct way to check if a character ch is a whitespace character is:

```
boolean isSpace = Character.isWhitespace(ch);
```

The Character.isWhitespace() method determines whether a character is a whitespace character (e.g., space, tab, newline).

Option B is incorrect because Character.isWhite() is not a valid method.

Option C is invalid because ch is a primitive type char, which does not have methods.

17. A student wants to convert a character ch to lowercase. He writes the following incorrect statement:

ch = Character.toLower(ch);

What will be the correct statement?

A. ch = Character.toLowerCase(ch);

B. ch = Character.toLower(ch);

C. ch = ch.toLowerCase();

D. Both A and C

(a) Only A

(b) Only B

(c) Both A and C

(d) All three

Ans : MAIN 2009

(a) Only A

The correct way to convert a character ch to lowercase is:

```
ch = Character.toLowerCase(ch);
```

(i) Character.toLowerCase() is the correct method to convert a character to lowercase.

(ii) Option B is invalid because Character.toLower() does not exist in Java.

(iii) Option C is invalid because ch is a primitive type char and does not have methods like toLowerCase().

Downloaded from www.pdf.tube

```
int n;
// Here, 4 bytes = 4 × 8 = 32 bits
are reserved.
```

Example of composite data type:

```
java
class Complex {
    int a;
    double b;
}
```

Here, a requires 32 bits, and b requires 64 bits. So, composite type Complex requires 32 + 64 = 96 bits for each of its objects.

55. Name all possible members of a class.

Ans : MAIN 2000

Members of a class can be :

(i) **Data Members :** Also called member variables or instance variables.

(ii) **Constructors :** Functions used to initialize values to the member variables.

(iii) **Methods :** Also called member functions. They are used to apply a process on the object's attributes.

56. Can any primitive type be a class member variable?

Ans : MAIN 2023

Yes, primitive types can be class member variables in Java. These variables store simple values such as integers, floating-point numbers, characters, or boolean values. Primitive type member variables are defined inside a class and can be accessed and manipulated using instances of that class.

57. How can a class type variable support the accessing of members of a class?

Ans : SQP 2022

Class members can be accessed through class type variables (objects) by using the dot (.) operator.
e.g.,

```
object.memberFunction();
```

58. Can another class be a member of a class?

Ans : COMP 2021

Yes, another class can be a member of a class in Java. This is achieved through nested classes or inner classes. A nested class is a class defined within the body of another class. It is a member of its enclosing class and has access to its private members.

FIFTEEN MARKS QUESTIONS

59. Define a Product class with the following specifications:
Data Members (Instance Variables):
(i) productName (to store the name of the product)
(ii) price (to store the price of the product)
(iii) quantity (to store the quantity of the product available)
(iv) totalCost (to store the total cost of the product).
Member Methods:
(i) A parameterized constructor to initialize the data members.
(ii) A method to accept the details of a product.
(iii) A method to compute the total cost (price × quantity).
(iv) A method to display the product name, price, quantity, and total cost.
Write a main method to create an object of the Product class and call the above member methods.

Ans : MAIN 2023

```
import java.util.Scanner;

class Product {
    // Data members
    String productName;
    double price;
    int quantity;
    double totalCost;

    // Parameterized constructor
   public Product(String productName,
double price, int quantity) {
      this.productName = productName;
        this.price = price;
        this.quantity = quantity;
    }

      // Method to accept product
details
    public void acceptDetails() {
            Scanner scanner = new
Scanner(System.in);
            System.out.print("Enter
product name: ");
           productName = scanner.
nextLine();
            System.out.print("Enter
product price: ");
       price = scanner.nextDouble();
```

NODIA

CHAPTER 6

ENCAPSULATION

SUMMARY

1. ENCAPSULATION

This feature of object-oriented programming, where the data members and the methods are bound in a single unit and wrapped by a class, is called encapsulation. The data members visibility can be checked from being accessed outside the class.

2. VISIBILITY MODIFIERS/ACCESS MODIFIERS

These are the keywords that precede the declaration of variables or methods. They restrict the accessibility of the variables. Java provides four kinds of access:

2.1 Public

It allows the variables (data members) and methods visibility or accessibility to all classes outside the class in which it is defined, and also in other packages.
Syntax:

```
public datatype variable name;
public returntype method-name();
```

Example:

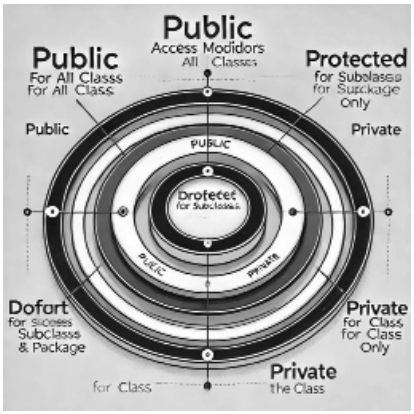
```
Example: public int a, b, c;

public void sum(int a, int b) {
    int c = a + b;
    System.out.print(c);
}
```

Protected, Default, and Private access levels are also explained further in the actual content.

2.2 Diagram

There is a diagram illustrating the relation between all access modifiers:
Public
Protected
Default
Private

It shows overlapping scopes such as within the class, package, subclasses, and all classes.



2.3 Protected

The protected variables and methods are accessible in the class in which they are defined and subclasses in the same package, as well as subclasses in other packages.
Example:

```
protected String x;
```

2.4 Private

The private members are accessible only within the class in which they are defined. They cannot be inherited to derived classes.
Example:

```
private int sum;
```

2.5 Friendly Access

When the declaration of members is not preceded by any access modifier. In the default access, these members can be accessed in all classes within the same package (note the difference from public access).

3. NON-ACCESS MODIFIERS

Non-access modifiers do not change the accessibility of variables and methods but do provide them special properties. They are:

Click Here to Buy book on Amazon

```
      System.out.println("Name: " +
name + "\t" + "Age: " + age);
    System.out.println("M1: " + m1 +
"\t" + "M2: " + m2 + "\t" + "M3: "
+ m3);
         System.out.println("Maximum
marks: " + maximum);
         System.out.println("Average
marks: " + average);
}

public static void main() {
   Student obj = new Student("Amber",
15, 85.0, 88.0, 84.0);
    obj.accept();
    obj.compute();
    obj.display();
}
}
```

Variables used :

name: Student's name

age: Student's age

m1, m2, m3: Marks in 3 subjects

maximum: Maximum marks

obj: Student object

61. Define a class Teacher described as below :

Data Members : Name, Address, Phone number, Highest Qualification, Subject Specialization, Monthly Salary, Income Tax.

Member Methods :

(i) To accept the details of a teacher.

(ii) To display the teacher's details.

(iii) To increment the monthly salary by 5000.

(iv) To compute the annual income tax as 5% of the annual salary above 150000.

Write a main method to create objects of the class and call the member methods.

Ans : COMP 2022

```
import java.util.*;
public class Teacher {
     String name, add, phno, qual,
subject;
    double salary, incometax;

    void Accept() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the name");
        name = buf.nextLine();
          System.out.println("Please
enter the address");
        add = buf.nextLine();
          System.out.println("Please
enter the phno");
        phno = buf.nextLine();
          System.out.println("Please
enter the qualification");
        qual = buf.nextLine();
          System.out.println("Please
enter the subject");
        subject = buf.nextLine();
    }
    subject = buf.nextLine();
    System.out.println("Please enter
the monthly salary");
    salary = buf.nextInt();
    }

    void display() {
      System.out.println("DETAILS");
         System.out.println("Name: "
+ name);
        System.out.println("Address:
" + add);
           System.out.println("Phone
number: " + phno);
        System.out.println("Subject:
" + subject);
         System.out.println("Highest
qualification: " + qual);
         System.out.println("Subject
Specialisation: " + subject);
         System.out.println("Salary:
" + salary);
          System.out.println("Income
Tax: " + incometax);
    }

    void increment() {
        salary = salary + 5000;
    }

    void compute() {
        double an = 12 * salary;
       incometax = 5.0 / 100.0 * (an
- 150000);
    }

     public static void main(String
a[]) {
        Teacher obj = new Teacher();
        obj.Accept();
        obj.increment();
```

NODIA

```
            System.out.print("Enter
product quantity: ");
        quantity = scanner.nextInt();
    }

    // Method to compute total cost
    public void computeTotalCost() {
        totalCost = price * quantity;
    }
      // Method to display product
details
    public void display() {
       System.out.println("\nProduct
Details:");
         System.out.println("Product
Name: " + productName);
        System.out.println("Price: $"
+ price);
        System.out.println("Quantity:
" + quantity);
           System.out.println("Total
Cost: $" + totalCost);
    }
    // Main method
    public static void main(String[]
args) {
        // Creating an object of the
Product class
            Product product = new
Product("", 0.0, 0);
        // Accepting product details
        product.acceptDetails();
        // Computing total cost
        product.computeTotalCost();
       // Displaying product details
        product.display();
    }
}
```

60. Define a Student class with the following specifications:
Data members/instance variables:
name, age, m1, m2, m3 (for marks in 3 subjects), maximum and average marks.
Member methods:
(i) A parameterized constructor to initialize the data members.
(ii) To accept the details of a student.
(iii) To compute the average and the maximum marks out of three marks.
(iv) To display the name, age, marks in three subjects, maximum and average marks.
Write a main method to create an object of a class and call the above member methods.

Ans : MAIN 2024

```
import java.util.*;
class Student {
    String name;
    int age;
      double m1, m2, m3, maximum,
average;

     // Parameterized constructor to
initialize data members
     Student(String n, int a, double
x, double y, double z) {S
        name = n;
        age = a;
        m1 = x;
        m2 = y;
        m3 = z;
    }
    // Method to accept details of a
student
    void accept() {
      Scanner b = new Scanner(System.
in);
          System.out.println("Enter
name and age");
        int ag = b.nextInt();
          System.out.println("Enter
marks in 3 subjects");
        m1 = b.nextDouble();
        m2 = b.nextDouble();
        m3 = b.nextDouble();
    }

    // Method to compute the average
and maximum marks
    void compute() {
        if (m1 > m2 && m1 > m3)
            maximum = m1;
        else if (m2 > m1 && m2 > m3)
            maximum = m2;
        else if (m3 > m1 && m3 > m2)
            maximum = m3;
        else
            // Marks may be equal
            System.out.print("Please
enter distinct marks");

       average = (m1 + m2 + m3) / 3.0;
    }
}
void display() {
       System.out.println("Student's
Data:");
```

Downloaded from www.pdf.tube

64. (i) Design a class Date with 3 data members: day, month, and year.
Member Methods :
(a) Accept date
(b) Display the date
(ii) Define main method in the class to invoke the defined methods for two dates.

Ans : MAIN 2019

```
import java.util.*;
public class Date {
    int day, month, year;

    void accept() {
     Scanner buf = new Scanner(System.
in);
           System.out.println("Enter
the date in form of day, month, and
year:");
        day = buf.nextInt();
        month = buf.nextInt();
        year = buf.nextInt();
    }
    void display() {
         System.out.println("The day
is: " + day);
       System.out.println("The month
is: " + month);
        System.out.println("The year
is: " + year);
    }
}
    public static void main() {
        Date dt = new Date();
        dt.accept();
        dt.display();
    }
}
```

65. WAP in Java that defines class Table whose description is given below :
Data Members (Instance variables) : length, breadth, and height in meters.
Member Methods :
(i) To accept the values
(ii) To display the values
(iii) Convert meters to centimeters.
Define main method to call the above methods.

Ans : SQP 2019

```
import java.util.*;
public class Table {
    int length, breadth, height;  //
Data Members
    void accept() {
     Scanner buf = new Scanner(System.
in);
        length = buf.nextInt();
        breadth = buf.nextInt();
        height = buf.nextInt();
    }
    void display() {
         System.out.println("Length=
" + length);
        System.out.println("Breadth=
" + breadth);
         System.out.println("Height=
" + height);
        convert();
    }
    void convert() {
        int cmlen = length * 100;
        int cmbred = breadth * 100;
        int cmht = height * 100;
       System.out.println("Length in
cms= " + cmlen);
         System.out.println("Breadth
in cms= " + cmbred);
       System.out.println("Height in
cms= " + cmht);
    }
     public static void main(String
args[]) {
        Table ob = new Table();
                BufferedReader buf
=       new       BufferedReader(new
InputStreamReader(System.in));
          System.out.println("1. To
input or enter values for Length,
breadth and height");
          System.out.println("2. To
display the values");
          System.out.println("3. To
convert and display the converted
value");
    }
}
          System.out.println("Enter
choice");
        int ch = buf.nextInt();
        if (ch == 1) ob.accept();
      else if (ch == 2) ob.display();
      else if (ch == 3) ob.convert();
        else System.out.println("Not
a valid choice");
    }
}
```

NODIA

```
        obj.compute();
        obj.display();
    }
}
```

62. Write a program to design a class Box with members as follows :
Data Members : length, breadth and height.
Member Methods :
(i) Compute volume of the box.
(ii) Compute the total area of the box.

Ans : SQP 2021

```
import java.util.*;
public class Box {
    int l, b, h;

    void Accept() {
     Scanner buf = new Scanner(System.
in);
        System.out.println("Enter the
length, breadth, and height");
        l = buf.nextInt();
        b = buf.nextInt();
        h = buf.nextInt();
    }

    void volume() {
        int vol = l * b * h;
       System.out.println("VOLUME OF
THE BOX: " + vol);
    }

    void area() {
        int a = 2 * (l * b + b * h +
l * h);
         System.out.println("Area of
box: " + a);
    }

    public static void main() {
        Box ob = new Box();
        ob.Accept();
        ob.volume();
        ob.area();
    }
}
```

63. (i) Define a class to represent your friends with the following details :
Data Members : serial number , name , address phone number , date of birth , birthday ,month ,birth year
Member Methods :
(a) To accept the information of a friend.
(b) To display a friend's details.
(ii) Write main to create class objects and call the methods.

Ans : MAIN 2020

```
import java.util.*;
public class Friend {
     int sno, day, month, year; //
Data members
    String name, add, phno;
    void accept() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the serial number, name, add,
phno, day, month, year:");
        sno = buf.nextInt();
        name = buf.nextLine();
        add = buf.nextLine();
        phno = buf.nextLine();
        day = buf.nextInt();
        month = buf.nextInt();
        year = buf.nextInt();
    }

    void display() {
          System.out.println("FRIEND
DETAILS");
        System.out.println();
          System.out.println("Serial
number: " + sno);
         System.out.println("Name: "
+ name);
        System.out.println("Address:
" + add);
           System.out.println("Phone
number: " + phno);
         System.out.println("Date of
Birth: " + day);
        System.out.println("Month: "
+ month);
           System.out.println("Birth
year: " + year);
    }
    public static void main() {
        Friend ob = new Friend();
        ob.accept();
        ob.display();
    }
}
```

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

68. Write a program to define a class Book, which is described as
Member Variables : Name , Author , Publication Cost ,Number of pages
Member Methods :
(i) Accept book details
(ii) Display the details
(iii) Increase the cost of the book by 50

Ans : MAIN 2017

```
import java.util.*;
public class Book {
    String nam, author, publication;
    double cost;
    int num;

    void accept() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the name of the book");
        nam = buf.nextLine();
          System.out.println("Please
enter the author name");
        author = buf.nextLine();
          System.out.println("Please
enter the publication name");
       publication = buf.nextLine();
          System.out.println("Please
enter the cost of the book, no. of
pages");
        cost = buf.nextDouble();
        num = buf.nextInt();
    }

    void display() {
         System.out.println("Name of
the book: " + nam);
         System.out.println("Name of
the author: " + author);
         System.out.println("Name of
the publication: " + publication);
         System.out.println("Cost of
the book: " + cost);
          System.out.println("No. of
pages: " + num);
    }

    void increment() {
        cost += 50;
    }

     public static void main(String
args[]) {
        Book bo = new Book();
        bo.accept();
        bo.display();
         System.out.println("Changed
values");
        bo.increment();
        bo.display();
    }
}
```

69. Write a Java program that defines a class Time as :
Data Members : hours, minutes, seconds.
Member Methods :
(i) Initialize values
(ii) Display the time
(iii) Compute time interval in two time objects and print it
(iv) Compute sum of the time objects and print it

Ans : COMP 2016

```
import java.util.*;
public class Time {
    int hrs, min, sec;

    void accept() {
     Scanner buf = new Scanner(System.
in);
     System.out.println("Please enter
the Hours, Minutes, and Seconds");
        hrs = buf.nextInt();
        min = buf.nextInt();
        sec = buf.venextInt();
    }

    void display() {
        System.out.println("The time
is: " + hrs + " hrs " + min + "
minutes and " + sec + " seconds");
    }

   void interval(Time t1, Time t2) {
         int totSeconds1 = t1.hrs *
3600 + t1.min * 60 + t1.sec;
         int totSeconds2 = t2.hrs *
3600 + t2.min * 60 + t2.sec;
            int in = totSeconds1 -
totSeconds2;
        int Hour = in / 3600;
       int minute = (in % 3600) / 60;
       int second = (in % 3600) % 60;
            System.out.println("Time
interval:");
        System.out.println("Hours: "
+ Hour + " hrs " + minute + " minutes
```

Downloaded from www.pdf.tube

```
and " + second + " seconds");
    }

    void sum(Time t1, Time t2) {
        int h, m, s, mm, ss;
        ss = (t1.sec + t2.sec) / 60;
         mm = (ss + t1.min + t2.min)
/ 60;
          m = (ss + t1.min + t2.min)
% 60;
        h = mm + t1.hrs + t2.hrs;
        if (h == 24) h = 0;
       System.out.println("Sum of the
two time objects is: " + h + " hrs "
+ m + " minutes " + s + " seconds");
    }

     public static void main(String
args[]) {
        Time ob1 = new Time();
        Time ob2 = new Time();
        Time ob = new Time();
        ob1.accept();
        System.out.println(ob1.hrs);
        ob2.accept();
        System.out.println(ob2.hrs);
        ob.interval(ob1, ob2);
        ob.sum(ob1, ob2);
    }
}
```

70. Write a program that accepts a character representing a hexadecimal digit, converts it to its equivalent decimal value, and prints the result. Demonstrate the use of auto-boxing and unboxing in the process.

Ans : MAIN 2017

```
class Convert {
    public static void main() {
            Character c = 'a';   //
Autoboxing
       char ch = c;        // Unboxing
        int d;
       if (ch >= '0' && ch <= '9') {
            d = ch - '0';  // Digits
0 to 9 remain as 0 to 9
         } else if (ch >= 'A' && ch
<= 'F') {
            d = ch - 'A' + 10;
         } else if (ch >= 'a' && ch
<= 'f') {
            d = ch - 'a' + 10;
        } else {
            System.out.print("Invalid
hexa character");
             System.exit(0);
         }
       System.out.print("Hexadecimal
character in decimal no.: " + d);
    }
}
```

71. Write a program that takes a character representing a binary digit, converts it to its equivalent decimal value, and prints the result. Demonstrate the use of auto-boxing and unboxing in the implementation.

Ans : COMP 2014

```
public class BinaryToDecimal {
    public static void main(String[]
args) {
     // Input: Character representing
a binary digit
        char binaryChar = '1';

      // Autoboxing: Converting char
to Character object
         Character binaryCharacter =
binaryChar;

            // Unboxing: Converting
Character to primitive char
          char unboxedChar = binary-
character;

       // Converting binary character
to its decimal equivalent
        int decimalValue = Character.
getNumericValue(unboxedChar);

        // Output the result
          System.out.println("Binary
Character: " + binaryChar);
         System.out.println("Decimal
Equivalent: " + decimalValue);
    }
}
```

72. Write a program that takes a character as an input and checks if the character is a vowel or a consonant. Allow the user to enter repeatedly until the user enters a digit as the character. Finally, print the number of vowels and the number of consonants.

Ans : SQP 2013

NODIA

```
import java.util.Scanner;
class Main {
     public static void main(String
args[]) {
          System.out.print("Enter a
character: ");
     Scanner sc = new Scanner(System.
in);
       char c = sc.next().charAt(0);
        int d = 0, e = 0;
            boolean x = Character.
isDigit(c);

        while (!x) {
                char f = Character.
toLowerCase(c);
            if (f == 'a' || f == 'e'
|| f == 'i' || f == 'o' || f == 'u') {
                System.out.println(f
+ " is a vowel");
                d = d + 1;
            } else if (f >= 'a' && f
<= 'z') {
                System.out.println(f
+ " is a consonant");
                e = e + 1;
            }
             System.out.print("Enter
a character: ");
            c = sc.next().charAt(0);
           x = Character.isDigit(c);
        }

       System.out.println("Number of
vowels: " + d);
       System.out.println("Number of
consonants: " + e);
    }
}
        System.out.println("To check
more characters, enter a character.\n
To exit, enter a digit.");

        c = sc.next().charAt(0);
        x = Character.isDigit(c);
    } // while loop body ends here
    System.out.print("No. of vowels:
" + d);
         System.out.print("\nNo.  of
consonants: " + e);
    }
}
```

Download INDIA's Best Study app NODIA

73. Write a program to input a sentence and convert it into uppercase and count and display the total number of words starting with a letter 'A'.
Example :
Example Input : ADVANCEMENT AND APPLICATION OF INFORMATION TECHNOLOGY ARE EVER CHANGING
Sample Output : Total number of words starting with letter 'A' = 4

Ans : MAIN 2012

```
import java.util.Scanner;
class Main {
     public static void main(String
args[]) {
          System.out.print("Enter a
sentence: ");
     Scanner sc = new Scanner(System.
in);
        String s = sc.nextLine();
       String str = s.toUpperCase();
        System.out.print("String is:
" + str);
      str = " " + str; // Concatenate
space in beginning
        int e = 0;
          for (int i = 0; i <= str.
length() - 2; i++) {
            char c = str.charAt(i);
          char d = str.charAt(i + 1);
       if (Character.isWhitespace(c)
== true && Character.isWhitespace(d)
== false) {
                if (d == 'A')
                    e = e + 1;
            }
        }
           System.out.print("No. of
words starting with A are: " + e);
    }
}
```

www.nodia.in

Click Here to Buy book on Amazon

66. Write a program to create a class Reserve Ticket with the following specifications :
Data Members : Name of the buyer ,Amount of the ticket ,Counter to count sold tickets
Member Methods :
(i) Accept values
(ii) Display the values
(iii) Increase ticket amount by 10

Ans : COMP 2017

```
import java.util.*;
public class ReserveTicket {
    String name;
    double amount;

    void accept() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the name of the buyer: ");
        name = buf.nextLine();

          System.out.println("Please
enter the amount of the ticket: ");
        amount = buf.nextDouble();
    }

    void display() {
         System.out.println("Name of
buyer: " + name);
       System.out.println("Amount of
ticket: " + amount);
    }

    void Increment() {
        amount += 10;
    }

     public static void main(String
args[]) {
           ReserveTicket obj = new
ReserveTicket();
        obj.accept();
        obj.display();

        obj.Increment();
         System.out.println("Changed
amount: ");
        obj.display();
    }
}
```

Download INDIA's Best Study app NODIA

NODIA

67. Create a class Donor with the following specifications :
Data Members : Donor's name Blood group Haemoglobin level (in percentage)
Member Methods :
(i) Accept the details of the donor
(ii) Display the donor details
Write a main method to call the above functions for a donor.

Ans : SQP 2016

```
import java.util.*;
public class Donor {
    String name, group;
    double hem;

    void accept() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the name of the donor");
        name = buf.nextLine();

          System.out.println("Please
enter the blood group of the donor");
        group = buf.nextLine();
          System.out.println("Please
enter  the  haemoglobin  level  in
percent:");
        hem = buf.nextDouble();
    }
    void display() {
         System.out.println("Name of
donor: " + name);
          System.out.println("Blood
group of donor: " + group);
            System.out.println("The
haemoglobin  level  in  percent:  " +
hem + "%");
    }

     public static void main(String
args[]) {
        Donor ob = new Donor();
        ob.accept();
        ob.display();
    }
}
```

Download INDIA's Best Study app NODIA

static
final
abstract

4. SCOPE OF VARIABLES

Scope refers to the region where the variable can be accessed, depending on the part of the program where they are declared. Depending on the scope, they are classified as:

1. Local Variables
2. Instance Variables
3. Class Variables

4.1 Local Variables

Variables declared and used within a method are called local variables. They cannot be used outside the body of the method.
Example:

```
void sum() {
  int a = 5, b = 10, c;
  c = a + b;
  System.out.print(c);
}
```

Here, the scope of a, b, and c is the method sum(); hence, they are local variables.

4.2 Instance Variables / Member Variables

These are declared in the class body but are not contained in any constructor or method.
Example:

```
class Abc {
  int num1;
  double num2;

  Abc() {
    num1 = 0;
    num2 = 0;
  }
}
```

Here, num1 and num2 are instance variables. They may take different values for each object of the class as, whenever an object is created, a new copy of a memory location (instance variable) is created.

4.3 Argument Variables

An argument variable is an actual parameter used when a function is called. Its value is passed to a formal parameter.
Example:
To call a function sum which returns an integer and takes two integers as parameters, the following calling statement is written:

```
int x = 7, y = 9;
int s = sum(x, y);
```

Here, x and y are argument variables.

4.4 Class Variables

These variables are associated with the class rather than its objects. The values they take are global to the class and do not belong to any object. They are created by preceding the variable declaration statement with the keyword static.
Example:

```
class Abc {
  static int x, y;   // x and y are
class variables here
  static void sum() {
    // statements;
  }
}
```

MULTIPLE CHOICE QUESTION

1. Which of the following access control must be used for the main method to be invoked by the JVM?
(a) Private
(b) Public
(c) Protected
(d) No access

Ans : SQP 2024

(b) Public
The main method in Java must be declared as public for the Java Virtual Machine (JVM) to invoke it. This is because the main method serves as the entry point of a program and needs to be accessible globally. If declared as private, protected, or without access modifiers (default), the JVM cannot access it, leading to a runtime error.

2. Which of the following keywords helps access a class member without creating an object and without associating a variable with an object?
(a) Private
(b) Final
(c) Static
(d) Protected

Ans : COMP 2023

Downloaded from www.pdf.tube

(a) (2) (3) (1) (4)
(b) (1) (3) (2) (4)
(c) (2) (1) (3) (4)
(d) (3) (2) (1) (4)

Ans : COMP 2002

(c) (2) (1) (3) (4)
Heres how it works step by step:

(i) Condition Check: The condition (x > y) is evaluated.
  (a) If x > y, it means x is the larger value.
  (b) If the condition is false, y is the larger value.

(ii) Assignment: Based on the condition, the larger value is assigned to max.

(iii) Return (if applicable): Finally, the value of max is returned or used in further operations.

This logical flow ensures that the larger of the two values is correctly assigned to the max variable and returned or used as needed. It is simple, efficient, and easy to understand.

43. Consider the following program segment where the statements are jumbled. Choose the correct order of statements to reverse a number.

```
void reverseNumber(int num) {
    reversed = reversed * 10 + digit;
// (1)
    digit = num % 10;                    //
(2)
    num = num / 10;                      //
(3)
    int reversed = 0;                    //
(4)
}
```

(a) (4) (2) (1) (3)
(b) (4) (2) (3) (1)
(c) (3) (4) (2) (1)
(d) (4) (3) (2) (1)

Ans : MAIN 2003

(a) (4) (2) (1) (3)
Initialize reversed: Start with reversed = 0 to store the reversed digits.
Extract the last digit: Use num % 10 to get the last digit of the number.
Update reversed: Multiply reversed by 10 and add the extracted digit to shift digits and append the new one.
Reduce num: Use integer division num / 10 to remove the last digit.

44. Consider the following program segment where the statements are jumbled. Choose the correct order of statements to calculate the factorial of a number.

```
void factorial(int n) {
    result = result * i;  // (1)
     for (int i = 1; i <= n; i++) {
// (2)
    int result = 1;          // (3)
    return result;           // (4)
}
```

(a) (3) (2) (1) (4)
(b) (1) (2) (3) (4)
(c) (3) (4) (1) (2)
(d) (4) (3) (2) (1)

Ans : SQP 2004

(a) (3) (2) (1) (4)
Initialize result, iterate through i to multiply, and return the result.

Initialize result: Start with result = 1 to store the product of numbers.

Iterate through i: Use a loop to multiply result by each integer from 1 to n.

Return the result: After the loop, result holds the factorial value, which is returned.

Start with result = 1 because the factorial of 0 or 1 is 1.

Multiply result by each number from 1 to n.

Return the final product as the factorial of n.

45. Consider the following program segment where the statements are jumbled. Choose the correct order of statements to check if a number is even or odd.

```
void checkEvenOdd(int num) {
 if (num % 2 == 0)  // (1)
  System.out.println("Even");  // (2)
elseSystem.out.println("Odd");// (3)
   return;  // (4)
}
```

(a) (1) (3) (2) (4)
(b) (1) (2) (3) (4)
(c) (3) (1) (2) (4)
(d) (2) (3) (1) (4)

Ans : COMP 2005

(b) (1) (2) (3) (4)
Check the condition: Use the modulus operator % to check if the number is divisible by 2 (num % 2 == 0).

NODIA

Ans : COMP 2013

(b) protected
The protected access modifier allows variables and methods to be accessed by subclasses, even if they are in different packages. This is in addition to access within the same package. In contrast, public provides universal access, private restricts access to the same class, and default limits access to the same package only.

9. What is an instance variable?
(a) A variable declared with the static keyword
(b) A variable declared inside a method
(c) A variable declared in the class body and unique to each object
(d) A parameter passed to a method

Ans : SQP 2011

(c) A variable declared in the class body and unique to each object
An instance variable is a variable declared in the class body but outside any method, constructor, or block. It is associated with an object of the class, meaning each object gets its own copy of the instance variable. These variables hold values unique to each instance of the class and are not shared among objects unless specified otherwise (e.g., static variables).

10. Which keyword is used to declare a constant variable in Java?
(a) Static
(b) Abstract
(c) Final
(d) Default

Ans : COMP 2010

(c) Final
In Java, the final keyword is used to declare a constant variable. A variable declared as final cannot have its value modified after it is initialized. It is commonly used to define constants in a program. static is used for class-level variables, abstract is used for methods and classes, and default is used for default methods in interfaces.

11. What happens if a method declares a variable as local?
(a) It is shared among all objects of the class.
(b) It is accessible only within the method where it is declared.
(c) It is accessible by subclasses in other packages.
(d) It cannot be initialized.

Ans : MAIN 2008

(b) It is accessible only within the method where it is declared.
A local variable is declared within a method, constructor, or block and is only accessible within that specific method or block. It is created when the method is called and destroyed when the method execution ends. Local variables are not shared among objects or accessible outside the method. They must be initialized before use.

12. Which access modifier allows a variable to be accessed from any class or package?
(a) Private
(b) Protected
(c) Public
(d) Default

Ans : SQP 2009

(c) Public
The public access modifier allows a variable or method to be accessed from any class or package. It provides the highest level of accessibility in Java. Variables or methods declared as public can be accessed universally. In contrast, private restricts access to the same class, protected allows access in the same package and subclasses, and default limits access to the same package only.

13. What type of variable is shared among all objects of a class?
(a) Local Variable
(b) Instance Variable
(c) Argument Variable
(d) Class Variable

Ans : COMP 2006

(d) Class Variable
A class variable is declared using the static keyword in Java. It is shared among all objects of the class, meaning all instances of the class refer to the same memory location for this variable. In contrast, instance variables are unique to each object, local variables are confined to methods or blocks, and argument variables are temporary and exist only during method execution.

14. Which keyword is used to declare a variable or method as a class-level member?
(a) Final
(b) Static
(c) Abstract
(d) Private

Click Here to Buy book on Amazon

(c) Static
The static keyword in Java allows class members (variables or methods) to be accessed without creating an object of the class. Static members belong to the class itself rather than to any specific object, making them directly accessible using the class name. Private restricts access, Final prevents modification, and Protected controls inheritance-based access.

3. To access the class members from other classes in the same package (and not outside the package), which of the following specifiers should be used?
(a) Friendly/default access
(b) Private
(c) Public
(d) Protected

Ans : MAIN 2011

(a) Friendly/default access
In Java, the default access modifier (also known as "friendly" access) allows class members to be accessed by other classes within the same package but restricts access from classes outside the package. The private modifier limits access to within the same class, protected allows access to subclasses (even outside the package), and public allows universal access.

4. _____ Variables are the variables declared inside a method and have a scope limited to the method. They cannot be accessed outside the method.
(a) Static (b) Local
(c) Global (d) Protected

Ans : MAIN 2013

(b) Local
Local variables are declared inside a method, constructor, or block and are accessible only within that scope. They are created when the method starts and destroyed when it ends, ensuring their scope is limited to the method. Static and Global variables have wider scopes, and Protected is an access modifier, not related to variable scope inside a method.

5. Which of the below specifiers provide minimum security for the class members?
(a) Private
(b) Protected
(c) Public
(d) Default/friendly access

Ans : COMP 2012

(c) Public
The public access specifier provides the minimum security for class members because it allows them to be accessed from any other class, regardless of the package. In contrast, private provides the highest security (accessible only within the same class), followed by protected (accessible in the same package and subclasses), and default/friendly access (accessible only within the same package).

6. The access modifier that gives least accessibility is:
(a) Private
(b) Public
(c) Protected
(d) Package

Ans : SQP 2016

(a) Private
The private access modifier provides the least accessibility in Java. Members declared as private are accessible only within the class in which they are defined. They cannot be accessed from subclasses, other classes, or even other classes in the same package. In contrast, public offers the highest accessibility, protected allows access in the same package and subclasses, and default/package access is restricted to the same package.

7. What is the scope of a private variable?
(a) Within the class and its subclasses
(b) Only within the class in which it is defined
(c) Within the same package
(d) Accessible from any class

Ans : MAIN 2015

(b) Only within the class in which it is defined
The scope of a private variable is limited to the class in which it is declared. It cannot be accessed directly from outside the class, including from subclasses or other classes within the same package. To access private variables, methods like getters and setters must be used.

8. Which access modifier allows variables and methods to be accessed by subclasses even if they are in different packages?
(a) Public
(b) Protected
(c) Private
(d) Default

NODIA

Ans : MAIN 2001

(d) Friendly
If no modifier is specified, the default (friendly) access allows variables and methods to be accessed within the same package.

22. Which of the following is an example of a local variable?
(a) int x; declared in the class body
(b) int x; declared inside a method
(c) static int x;
(d) protected int x;

Ans : COMP 2006

(b) int x; declared inside a method
Local variables are declared within a method and are accessible only within the method's scope.

23. Which of these can access protected members of a class?
(a) Only the class in which they are declared
(b) Classes within the same package and subclasses
(c) Any class in any package
(d) Only subclasses in the same package

Ans : MAIN 2011

(b) Classes within the same package and subclasses
protected members are accessible in the declaring class, subclasses, and other classes within the same package.

24. What keyword is used to define class variables?
(a) Final
(b) Abstract
(c) Static
(d) Volatile

Ans : COMP 2015

(c) Static
Class variables are declared using the static keyword and belong to the class rather than any object.

25. Which type of variable is specific to each object of a class?
(a) Static variable
(b) Instance variable
(c) Local variable
(d) Global variable

Ans : SQP 2017

(b) Instance variable
Instance variables are unique to each object of a class, with each object maintaining its own copy.

26. What are argument variables?
(a) Variables declared inside a method
(b) Variables passed to a method when it is called
(c) Variables declared in a class body
(d) Variables that use the static keyword

Ans : MAIN 2019

(b) Variables passed to a method when it is called
Argument variables are actual parameters used in a function call and are passed to the function's formal parameters.

27. Which of these is a non-access modifier in Java?
(a) Public
(b) Private
(c) Final
(d) Protected

Ans : COMP 2021

(c) Final
final is a non-access modifier that makes variables constant, methods un-overridable, and classes non-inheritable.

28. What is the scope of a local variable?
(a) Throughout the program
(b) Within the method it is declared
(c) Within the class
(d) Within the package

Ans : SQP 2005

(b) Within the method it is declared
Local variables are limited to the method in which they are declared and cannot be accessed outside.

29. Which access modifier allows variables to be accessible only in the same package?
(a) Public
(b) Protected
(c) Default
(d) Private

Ans : MAIN 2017

(c) Default
The default access modifier (friendly) allows variables and methods to be accessed only within the same package.

NODIA
Download INDIA's Best Study app NODIA

Ans : MAIN 2009

(b) Static
The static keyword in Java is used to declare a variable or method as a class-level member. Static members belong to the class rather than to any specific instance, meaning they are shared among all objects of the class. In contrast, final is used to make variables constant, abstract is used for abstraction, and private controls access to members.

15. If no access modifier is specified for a variable, it is accessible:
(a) Within the same package only
(b) From any package
(c) Only within the same class
(d) From subclasses in any package

Ans : COMP 2005

(a) Within the same package only
If no access modifier is specified for a variable, it has default (or package-private) access. This means the variable is accessible only within the same package. It is not accessible from subclasses outside the package or from classes in other packages. For broader accessibility, explicit modifiers like public or protected must be used.

16. Which access modifier allows variables to be accessible in all classes, including other packages?
(a) Public
(b) Private
(c) Default
(d) Protected

Ans : SQP 2021

(a) Public
The public access modifier makes variables and methods accessible in all classes and packages.

17. Which modifier restricts accessibility to the same class only?
(a) Public
(b) Private
(c) Protected
(d) Default

Ans : COMP 2023

(b) Private
The private access modifier restricts variable and method access to the class in which they are declared.

18. Which of the following is a local variable?
(a) A variable declared inside a method and used only within that method
(b) A variable declared in the class body but outside any method
(c) A variable declared with the static keyword
(d) A variable declared with public and used across all classes

Ans : SQP 2004

(a) A variable declared inside a method and used only within that method
A local variable is declared within a method, constructor, or block and can only be used within that specific method or block. It is created when the method starts and destroyed when it ends. It is not accessible outside the method. In contrast, options (b), (c) and (d) describe class-level or globally accessible variables.

19. What is encapsulation in object-oriented programming?
(a) Binding methods and variables together in a class
(b) Defining methods in an interface
(c) Extending the functionality of a base class
(d) Restricting variable access within a function

Ans : COMP 2015

(a) Binding methods and variables together in a class
Encapsulation is the concept of bundling data members and methods into a single unit (class), while controlling access to data using access modifiers.

20. Which of these is NOT an access modifier in Java?
(a) Public
(b) Private
(c) Static
(d) Protected

Ans : MAIN 2019

(c) Static
Static is a non-access modifier in Java, providing special properties to variables and methods but not controlling accessibility.

21. What is the default access modifier in Java?
(a) Public
(b) Protected
(c) Private
(d) Friendly

Downloaded from www.pdf.tube

The statement z = *5; is incorrect syntax. The correct forms are:
z *= 5; (compact multiplication assignment)
z = z * 5; (explicit multiplication).
The answer here aligns with "Only A" based on the given context.

38. A student wrote the following code to divide the value of a variable x by 4:
`x = /4;`
Which of the following statements will be correct?
(i) x /= 4;
(ii) x = x / 4;
(iii) x = /4;
(a) Only A
(b) Only B
(c) Both A and B
(d) All three

Ans : SQP 2002

(b) Only B
The statement x = /4; is invalid syntax. The correct forms are:
x = x / 4; (explicit division).
This question appears to omit the compact form (x /= 4) in the answer context.

39. A programmer wants to increment the value of a by 1.
They wrote:
`a = +;`
Which statement is correct?
(i) a++;
(ii) a = a + 1;
(iii) a = +;
(a) Only A
(b) Only B
(c) Both A and B
(d) All three

Ans : COMP 2001

(a) Only A
a++; is a shorthand for incrementing a by 1. While a = a + 1; is also correct, the question emphasizes the most compact form.

40. A developer attempts to subtract 10 from a variable n using the statement:
`n = -10;`
Which of the following is correct?
(i) n -= 10;
(ii) n = n - 10;
(iii) n = -10;
(a) Only A
(b) Only B
(c) Both A and B
(d) All three

Ans : MAIN 2000

(b) Only B
- n = n - 10; is the explicit way to subtract 10 from the variable n, providing clear readability for beginners or when explaining code.
- n -= 10; is a shorthand assignment operator that achieves the same result but in a more concise manner.

41. A programmer needs to add 7 to the value of m.
They wrote:
`m = +7;`
Which of the following will correctly achieve this?
(i) m += 7;
(ii) m = m + 7;
(iii) m = +7;
(a) Only A
(b) Only B
(c) Both A and B
(d) All three

Ans : SQP 2001

(a) Only A
The shorthand form m += 7; is indeed the most compact and accurate way to achieve the desired operation of adding 7 to the current value of m. It combines the addition operation and assignment into a single concise statement.
For example:
```
m += 7;  // Equivalent to m = m + 7;
```
Advantage : Its both compact and clear, making it a preferred choice in professional coding practices.
Clarity : While explicit forms like m = m + 7; may be clearer for beginners, m += 7; is equally understandable with a little familiarity.
This shorthand is highly recommended for clean and efficient code.

42. Consider the following program segment where the statements are jumbled. Choose the correct order of statements to assign the larger of two numbers to a variable max using conditional statements.
```
void findMax(int x, int y) {
    max = y;          // (1)
    if (x > y)        // (2)
    max = x;          // (3)
    return max;       // (4)
}
```

NODIA

30. Which access modifier restricts variable inheritance to subclasses in the same or other packages?

(a) Private
(b) Public
(c) Protected
(d) Default

Ans : SQP 2019

(c) Protected
The protected modifier allows access in the class, subclasses, and same-package classes but not other package classes.

31. What happens when a variable is declared with static?

(a) It becomes an instance variable.
(b) It belongs to the class and not any specific object.
(c) It becomes private.
(d) It is removed from memory after the method ends.

Ans : MAIN 2000

(b) It belongs to the class and not any specific object.
A static variable is shared across all instances of the class and belongs to the class itself.

32. Can private variables be inherited by subclasses?

(a) Yes, but only in the same package.
(b) Yes, in all cases.
(c) No, they cannot be inherited.
(d) Yes, but only if declared static.

Ans : COMP 2005

(c) No, they cannot be inherited.
Private variables are not inherited and can only be accessed within the class where they are declared.

33. What type of variable is declared in the class body and is specific to each object?

(a) Local variable
(b) Argument variable
(c) Instance variable
(d) Static variable

Ans : MAIN 2009

(c) Instance variable
Instance variables are declared in the class body and are specific to each object of the class.

34. Which variable scope lasts only during the execution of a method?

(a) Static variables
(b) Local variables
(c) Instance variables
(d) Class variables

Ans : COMP 2011

(b) Local variables
Local variables exist only during the execution of the method in which they are declared.

35. Which access modifier has the widest scope?

(a) Private
(b) Protected
(c) Public
(d) Default

Ans : SQP 2014

(c) Public
The public modifier provides the widest scope, allowing access in all classes and packages.

36. What does the final modifier do when applied to a class?

(a) Makes it accessible only in the same package
(b) Prevents the class from being inherited
(c) Allows multiple inheritance
(d) Makes all variables in the class static

Ans : COMP 2018

(b) Prevents the class from being inherited
The final modifier prevents the class from being extended by any subclass.

37. A programmer wants to multiply the value of a variable z by 5.
They wrote the following statement, which is incorrect:

```
z = *5;
```

What will be the correct statement?
(i) z *= 5;
(ii) z = z * 5;
(iii) z = *5;

(a) Only A
(b) Only B
(c) Both A and B
(d) All three

Ans : MAIN 2003

(a) Only A

Click Here to Buy book on Amazon

Print "Even" or "Odd": Based on the condition, print "Even" if true or "Odd" if false.
Return: End the function after printing.
The condition num % 2 == 0 evaluates to true for even numbers and false for odd numbers.
Based on the result, print the appropriate message.
The method completes execution after the print statement. The return is optional if the methods return type is void.

46. Consider the following program segment where the statements are jumbled. Choose the correct order of statements to find the smallest of three numbers.

```
void findSmallest(int a, int b, int c) {
    smallest = c;            // (1)
    if (b < smallest)        // (2)
    smallest = b;            // (3)
    if (a < smallest)        // (4)
    smallest = a;            // (5)
    return smallest;         // (6)
}
```

(a) (1) (2) (4) (3) (5) (6)

(b) (1) (3) (2) (4) (5) (6)

(c) (1) (2) (3) (4) (5) (6)

(d) (3) (1) (2) (4) (5) (6)

Ans : SQP 2006

(c) (1) (2) (3) (4) (5) (6)
Initialize smallest: Start by assuming c is the smallest (smallest = c).

Compare with b: Check if b is smaller than smallest and update if true.

Compare with a: Check if a is smaller than smallest and update if true.

Return the result: After the comparisons, return the value of smallest.

Start by assuming c is the smallest.

Update smallest if b or a is found to be smaller during comparisons.

Return the final value of smallest, which holds the smallest number among a, b, and c.

47. **Assertion (A) :** Encapsulation is a mechanism that binds data members and methods into a single unit.
Reason (R) : Encapsulation helps in restricting the access of variables using access modifiers like private, protected, and public.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : MAIN 2007

(a) Both A and R are true, and R is the correct explanation of A.
Encapsulation indeed binds data and methods, forming a cohesive unit. Access modifiers (private, protected, public) are tools to control access, ensuring that encapsulation restricts access as needed. Thus, the reason is a correct explanation.

48. **Assertion (A) :** Variables declared with the private modifier can only be accessed within the class in which they are defined.
Reason (R) : Private variables are not accessible even in the subclasses of the class in which they are defined.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : SQP 2008

(a) Both A and R are true, and R is the correct explanation of A.
private ensures that variables or methods are accessible only within the same class. Subclasses cannot access private members directly, even if they inherit from the class. Thus, both the assertion and reason are correct, and R explains A.

49. **Assertion (A) :** A static variable is shared among all objects of the class.
Reason (R) : The static keyword makes the variable instance-specific, creating a separate copy for each object.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : COMP 2009

NODIA

(c) A is true, but R is false.
A static variable is class-level and shared by all objects, not instance-specific. The reason incorrectly claims that static creates separate copies for each object, contradicting the assertion.

50. **Assertion (A) :** A local variable is accessible only within the method or block in which it is declared.
Reason (R) : Local variables are initialized automatically when they are declared.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : SQP 2010

(c) A is true, but R is false.
Local variables are indeed method/block-specific, but they are not automatically initialized. They must be explicitly initialized before use. Hence, the reason is false.

51. **Assertion (A) :** The protected access modifier allows access to members within the same package and subclasses in other packages.
Reason (R) : Members with protected access cannot be accessed directly by unrelated classes outside the package.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : COMP 2011

(a) Both A and R are true, and R is the correct explanation of A.
The protected modifier allows access within the same package and to subclasses in other packages. Unrelated classes outside the package cannot access protected members. The reason is correct and provides an accurate explanation of the assertion.

Download INDIA's Best Study app NODIA

TWO MARKS QUESTIONS

52. Define encapsulation. Why is it important?

Ans : MAIN 2024

Encapsulation is the process of bundling data members and methods into a single unit, wrapped in a class. It ensures controlled access to data, improving security and reducing complexity by restricting direct access from outside the class.

53. List the four access modifiers provided by Java.

Ans : SQP 2023

The four access modifiers in Java are:
Public
Protected
Private
Default (Friendly Access)

54. What is the purpose of access modifiers?

Ans : COMP 2021

Access modifiers control the visibility and accessibility of variables and methods, ensuring appropriate levels of data protection and accessibility across classes and packages.

55. Explain the scope of local variables with an example.

Ans : MAIN 2018

Local variables are declared and used within a method and cannot be accessed outside it.
Example:

```
void sum() {
    int a = 5, b = 10, c;
    c = a + b;
    System.out.print(c);
}
```

Here, a, b, and c are local to the method sum().

56. What are instance variables? Give an example.

Ans : SQP 2019

Instance variables are declared in a class but outside any method or constructor. They are unique to each object.
Example:

```
class Abc {
    int num1;
    double num2;
}
```

Here, num1 and num2 are instance variables.

Downloaded from www.pdf.tube

57. Differentiate between private and public access modifiers.

Ans : COMP 2015

Private: Members are accessible only within the defining class.
Public: Members are accessible to all classes, including those in other packages.

58. What is a class variable? How is it declared?

Ans : MAIN 2017

Class variables are shared by all objects of a class and declared using the static keyword.
Example:

```
static int x;
```

59. Explain the scope of protected access modifiers.

Ans : SQP 2016

Protected members are accessible within the class, its subclasses, and other classes in the same package. They are also accessible in subclasses outside the package.

60. Distinguish between instance variables and class variables.

Ans : SQP 2013

The difference between instance variable and class variable are as follows :

Instance Variables	Class Variables
Instance variables are not common to all instances of a class.	Class variables are common to all instances of a class.
Instance variables belong to an object since an object is an instance of a class. Each object will maintain its own copy of instance variables.	It has only one copy of variable(s) that are shared to all.
For example: class ABC {int x;}	Class variables are declared with the keyword static. For example: class ABC {static int x = 5;}
Instance variables are declared without the static keyword.	

61. What are non-access modifiers? Give two examples.

Ans : COMP 2015

Non-access modifiers provide special properties to variables and methods without affecting their visibility.
Examples:
(i) Static
(ii) Final

62. What happens when no access modifier is specified for a variable?

Ans : MAIN 2014

When no access modifier is specified, the variable has default (friendly) access, meaning it is accessible only within classes in the same package.

63. Explain any two types of access specifiers.

or

Differentiate the private and protected modifiers.

Ans : MAIN 2012

In Java programming, the various access specifiers are public, protected, private, and default.
(i) Private : Allows a variable or function to be accessed only within the same class in which it is declared.
(ii) Protected : Allows access to the class members within the same package and subclasses in other packages.

64. State the Java concept that is implemented through:
(i) A superclass and a subclass
(ii) The act of representing essential features without including background details

Ans : SQP 2011

(i) Inheritance : The concept of object-oriented programming is implemented through subclasses (child classes) and superclass's (parent classes).
(ii) Abstraction : The feature of object-oriented programming refers to representing only essential features without including background details.

65. What does encapsulation lead to?

Ans : COMP 2010

Encapsulation leads to abstraction and inheritance. Abstraction refers to the act of representing only

NODIA

essential features and hiding unnecessary details. Inheritance leads to code re usability.

66. How is encapsulation useful in an object-oriented programming environment?

Ans : SQP 2009

Since encapsulation leads to abstraction, it provides security to the hidden details (variables and methods) which may be manipulated by the user, if not hidden.

67. What is the practical implementation of encapsulation?

Ans : MAIN 2012

In Java, encapsulation is practically implemented by defining a class. A class is an entity that binds the variables and methods in a single unit.

68. Give an example to illustrate how encapsulation results in data hiding (or abstraction).

Ans : COMP 2008

A class, say Stats, contains the attributes height, weight, and age (as variables) of the candidates appearing for an interview. It also contains methods that operate on these variables. These variables and attributes are insulated by the class from outside access.
Example:

```
class Stats {
private int height, weight, age;
    void show() {
         System.out.print(height + " 
" + weight + " " + age);
    }
}
```

69. What does a class encapsulate?

Ans : SQP 2007

A class encapsulates the data members (attributes/ features/member variables) and the behavior (member functions) of the objects, which it can create and perform on the data member.

70. What are visibility modifiers?

Ans : MAIN 2006

The Java keywords that are applied to variable declarations or methods, in order to control the access or visibility, are called visibility modifiers.

Download INDIA's Best Study app NODIA

71. Differentiate the public and private modifiers.

Ans : COMP 2004

Private : Private member variables/methods can be accessed only within the same class in which they are declared.
Public : Public members can be accessed in their own class, subclasses, and all other classes in the same package as well as other packages.

72. Differentiate the default access from the public access.

Ans : COMP 2005

Default/Friendly Access : When the declaration of members is not preceded by any access modifier (also called friendly access), then these members can be accessed in all classes within the same package.
Public Access : It allows the variables (data members) and methods visibility to all classes outside the class in which it is defined and also in other packages.

```
      Syntax:
public datatype variable name;
public returntype methodname();
```

73. Consider the following class:

```
public class myClass {
    public static int x = 3, y = 4;
    public int a = 2, b = 3;
}
```

(i) Name the variables for which each object of the class will have its own distinct copy.
(ii) Name the variables that are common to all objects within the class.

Ans : MAIN 2003

(i) Each object of the class that will have its own copy includes variables a and b.
(ii) Variables that are common to all objects within the class are x and y.

74. A class Transport states the following declaration:
private int weight;
Answer the following questions:
(i) Is the following statement valid, if written in a method compare() in class Transport:
weight = 150;
(ii) Modify the given statement, to allow weight to be accessed in a class ABC which is a subclass of Transport
(iii) Modify, the given statement, to allow weight to be accessed in a class MONTANA which is

Click Here to Buy book on Amazon

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

in the same package that contains Transport.

(iv) Modify the statement to make it accessible to a class in another package that does not contain Transport.

Ans : SQP 2002

(i) Yes, it is valid as a private variable can be accessed in any method of the same class where it is declared.

(ii) protected int weight;

(iii) public int weight; or int weight;

(iv) public int weight;

75. What do you understand by the scope and lifetime of variables?

Ans : COMP 2001

Scope and Lifetime of Variables : Variables can be declared in different parts of a program, such as within a class, a method, or a block (enclosed by { and }). The scope of a variable is determined by the region where it is declared.

(i) Local Scope : Variables declared inside a method or a block are known as local variables. These variables are accessible only within the method or block where they are defined. Once the scope of the method or block ends, the variables are destroyed, meaning their lifetime is limited to their scope.

(ii) Class Scope : Variables declared outside of any methods, but within a class, are said to have class scope. These variables persist as long as the class object exists. They can be accessed and used by all methods within the class.

```
Example: class Learn {
    int x, y; // Class scope

    void input() {
        int z;       // Local scope
        z = x + y;   // Valid access
(x and y are class variables)
    }

    void output() {
      // Cannot access z (z has local
scope limited to the input method)
        // Can access x and y (x and
y have class scope)
    }
}
```

Download INDIA's Best Study app NODIA

76. Differentiate a member variable from a class variable. Illustrate it through an example.

Ans : MAIN 2001

Member variables : These are declared in the class body but not in any constructor or method.
Class variables : These variables are associated with the class rather than objects.
Example :

```
class Differ {
    int x, y;              // Member
variable
    static int a, b;       // Class
variable

    void access() {
        Differ obj = new Differ();
        obj.x = 10;
        obj.y = 10;

         a = 1;              // Not
associated with objects
        b = 2;
    }
}
```

77. Differentiate the scope of global variables and local variables.

Ans : SQP 2000

Since Java is a purely OP language, the global scope of variables refers to the scope of member variables and class variables, which can be accessed in all methods of the class in which they are declared. In contrast, the scope of a local variable is confined to the method or block where it is declared.

78. Consider the following class declaration and answer the questions that follow:

```
class Abc {
      int c, a = 5, b = 10;   //
statement 1

    void sum() {
       int a = 100;      // statement
2
        c = a + b;
    }
}
```

Download INDIA's Best Study app NODIA

Downloaded from www.pdf.tube

```
VowelConsonantTransformer {
    // Method to check if a character
is a vowel
   public static boolean isVowel(char
ch) {
     return "AEIOUaeiou".indexOf(ch)
!= -1;
    }

     // Method to get the previous
uppercase vowel
             public   static   char
getPreviousVowel(char vowel) {
        switch (vowel) {
            case 'A': return 'U';
            case 'E': return 'A';
            case 'I': return 'E';
            case 'O': return 'I';
            case 'U': return 'O';
           default: return vowel; //
Return unchanged if not a vowel
        }
    }

    // Method to transform the string
           public   static   String
transformString(String input) {
         StringBuilder result = new
StringBuilder();

             for (char ch : input.
toCharArray()) {
       if (Character.isUpperCase(ch)
&& isVowel(ch)) {
                                result.
append(getPreviousVowel(ch));     //
Replace uppercase vowels
               } else if (Character.
isLowerCase(ch) && !isVowel(ch)) {
                 result.append((char)
(ch  +  1));  //  Replace  lowercase
consonants
            } else {
                   result.append(ch);
// Keep digits, symbols, and other
characters unchanged
            }
        }

        return result.toString();
    }

    public static void main(String[]
args) {
             Scanner  scanner  =  new
Scanner(System.in);

        // Accept input from the user
           System.out.print("Enter  a
string: ");
            String  input  =  scanner.
nextLine();

        // Transform the string
          String transformedString =
transformString(input);

        // Output the result
     System.out.println("Transformed
String: " + transformedString);

        scanner.close();
    }
}
```

86. Define a class to accept a number and check whether it is a PERFECTSPY number or not. A number is called PERFECTSPY if the product of its digits equals the number of digits.
Example 1:
Input: 112
Output:
PERFECTSPY number [PRODUCT OF THE DIGITS $= 1 \times 1 \times 2 = 2$, NUMBER OF DIGITS $= 2$]

Ans : SQP 2014

NODIA

```
import java.util.Scanner;

public class PerfectSpyNumber {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Accept the input number
           System.out.print("Enter  a
number: ");
        int num = sc.nextInt();

         // Initialize variables for
product and digit count
        int product = 1, count = 0;
        int temp = num;

       // Calculate product of digits
and count digits
```

```
        String input = sc.nextLine();

            // Convert the string to
uppercase
       String uppercaseInput = input.
toUpperCase();

            // Initialize the output
string
          StringBuilder output = new
StringBuilder();

        // Process each character in
the string
     for (int i = 0; i < uppercaseInput.
length(); i++) {
            char ch = uppercaseInput.
charAt(i);

            // Check if the character
is a vowel
             if (isVowel(ch)) {
              output.append((char) (ch
- 1)); // Replace with the previous
ASCII character
             }
            // Check if the character
is a consonant
                 else if (Character.
isLetter(ch)) {
                 output.append((char)
(ch + 1)); // Replace with the next
ASCII character
             }
           // Keep digits and special
characters as they are
             else {
                 output.append(ch);
             }
        }

          // Display the transformed
string
     System.out.println("Transformed
String: " + output);
    }

     // Helper method to check if a
character is a vowel
  private static boolean isVowel(char
ch) {
          return "AEIOU".indexOf(ch)
!= -1;
    }
}
```

82. Define a class to accept a string and convert it to lowercase. Create and display a new string by replacing each vowel with the next character in the ASCII sequence and each consonant with the previous character in the ASCII sequence. Any other characters (digits or symbols) should remain unchanged.
Example:
Input: Hello@123
Output: gdkkp@123

Ans : COMP 2016

```
import java.util.Scanner;

public class StringTransformer {
    // Method to check if a character
is a vowel
   public static boolean isVowel(char
ch) {
        return "aeiou".indexOf(ch) !=
-1; // Check for vowels in lowercase
    }

    // Method to transform the string
            public   static   String
transformString(String input) {
         StringBuilder result = new
StringBuilder();

             for (char ch : input.
toCharArray()) {
          if (Character.isLetter(ch))
{ // Check if the character is a
letter
                        char lowerCh
=   Character.toLowerCase(ch);   //
Convert to lowercase for consistency
              if (isVowel(lowerCh)) {
                 result.append((char)
(ch + 1)); // Replace vowels with the
next character
                } else {
                result.append((char)
(ch - 1)); // Replace consonants with
the previous character
                }
            } else {
               result.append(ch); //
Keep digits and special characters
unchanged
            }
```

Click Here to Buy book on Amazon

```
        }

        return result.toString();
    }

    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

        // Accept input from the user
            System.out.print("Enter a string: ");
            String input = scanner.nextLine();

        // Transform the string
          String transformedString = transformString(input);

        // Output the result
     System.out.println("Transformed String: " + transformedString);

        scanner.close();
    }
}
```

83. Define a class to accept a string and convert it to uppercase. Create and display a new string where each vowel is replaced by the next vowel in sequence (A → E, E → I, I → O, O → U, U → A) and each consonant is replaced by the next consonant in alphabetical order. Non-alphabetic characters remain unchanged.
Example:
Input: #HELLO@2024
Output: #HIMMP@2024

Ans : SQP 2017

```
import java.util.Scanner;

public class VowelConsonantReplace {
    public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);

        // Accept the input string
            System.out.print("Enter a string: ");
       String input = sc.nextLine();

          // Convert the string to uppercase
      String uppercaseInput = input.toUpperCase();

          // Initialize the output string
         StringBuilder output = new StringBuilder();

        // Process each character
    for (int i = 0; i < uppercaseInput.length(); i++) {
           char ch = uppercaseInput.charAt(i);

              if (isVowel(ch)) { // Replace with the next vowel
       output.append(nextVowel(ch));
              } else if (Character.isLetter(ch)) { // Replace with the next consonant
                                  output.append(nextConsonant(ch));
              } else { // Keep non-alphabetic characters unchanged
                output.append(ch);
            }
        }

          // Display the transformed string
     System.out.println("Transformed String: " + output);
    }

   // Check if a character is a vowel
  private static boolean isVowel(char ch) {
          return "AEIOU".indexOf(ch) != -1;
    }

   // Get the next vowel in sequence
   private static char nextVowel(char ch) {
        switch (ch) {
            case A: return E;
            case E: return I;
            case I: return O;
            case O: return U;
            case U: return A;
            default: return ch;
```

```
        }
    }

     // Get the next consonant in
alphabetical order
            private   static   char
nextConsonant(char ch) {
         char nextChar = (char) (ch
+ 1);
        while (isVowel(nextChar)) {
            nextChar++;
        }
        return nextChar;
    }
}
```

84. Define a class to accept a string and convert it to lowercase. Create and display a new string by replacing each vowel with the character * and each consonant with the character #. Any other characters (digits or symbols) should remain unchanged.
Example:
Input: JaVa@2024
Output: ###@2024

Ans : MAIN 2010

```
import java.util.Scanner;

public class VowelConsonantReplacer
{
    // Method to check if a character
is a vowel
   public static boolean isVowel(char
ch) {
          return "aeiou".indexOf(ch)
!= -1;
    }

    // Method to transform the string
            public   static   String
transformString(String input) {
         StringBuilder result = new
StringBuilder();

             for (char ch : input.
toCharArray()) {
         if (Character.isLetter(ch))
{ // Check if the character is a
letter
                        char lowerCh
=   Character.toLowerCase(ch);   //
Convert to lowercase
             if (isVowel(lowerCh)) {
                 result.append('*');
// Replace vowels with '*'
                } else {
                 result.append('#');
// Replace consonants with '#'
                }
            } else {
               result.append(ch); //
Keep digits and special characters
unchanged
            }
        }

        return result.toString();
    }

    public static void main(String[]
args) {
             Scanner scanner = new
Scanner(System.in);

        // Accept input from the user
           System.out.print("Enter a
string: ");
            String input = scanner.
nextLine();

        // Transform the string
          String transformedString =
transformString(input);

        // Output the result
     System.out.println("Transformed
String: " + transformedString);

        scanner.close();
    }
}
```

85. Define a class to accept a string and create a new string where each uppercase vowel is replaced by the previous vowel in the sequence (A → U, E → A, etc.) and each lowercase consonant is replaced by the next consonant in the ASCII sequence. All other characters (digits or symbols) should remain unchanged.
Example:
Input: EdUcAtIoN!
Output: AdUdBtJpN!

Ans : COMP 2012

```
import java.util.Scanner;

public                          class
```

Downloaded from www.pdf.tube

```
        Scanner sc = new Scanner(System.
in);

        // Accept the input number
           System.out.print("Enter a
number: ");
        int num = sc.nextInt();

         // Initialize variables for
sum and product
        int sum = 0, product = 1;
        int temp = num;

          // Calculate sum of digits
and product of digits
        while (temp > 0) {
            int digit = temp % 10;
            sum += digit;
            product *= digit;
            temp /= 10;
        }

        // Check if the square of the
sum equals the product
        if ((sum * sum) == product) {
        System.out.println("SQUARESPY
number [SUM OF DIGITS = " + sum + ",
SQUARE OF SUM = " + (sum * sum) +
                              ", PRODUCT
OF DIGITS = " + product + "]");
        } else {
           System.out.println("Not a
SQUARESPY number [SUM OF DIGITS = "
+ sum + ", SQUARE OF SUM = " + (sum
* sum) +
                              ", PRODUCT
OF DIGITS = " + product + "]");
        }
    }
}
```

89. Define a class to accept a number and check whether it is a REVSPY number or not. A number is called REVSPY if the sum of its digits equals the reverse of the number of digits.
Example 1:
Input: 132
Output:
REVSPY number [SUM OF DIGITS = 1 + 3 + 2 = 6, REVERSE OF NUMBER OF DIGITS = 3 → 3]

Ans : MAIN 2011

```
import java.util.Scanner;

public class RevSpyNumber {

    public static void main(String[]
args) {
      Scanner sc = new Scanner(System.
in);

        // Accept the input number
           System.out.print("Enter a
number: ");
        int num = sc.nextInt();
// Initialize variables for sum of
digits and digit count
        int sum = 0, count = 0;
        int temp = num;
         // Calculate sum of digits
and count of digits
        while (temp > 0) {
            int digit = temp % 10;
            sum += digit;
            count++;
            temp /= 10;
        }
           // Reverse the number of
digits
                int reverseCount =
reverseNumber(count);
       // Check if the sum of digits
equals the reverse of the count
        if (sum == reverseCount) {
          System.out.println("REVSPY
number [SUM OF DIGITS = " + sum + ",
REVERSE OF NUMBER OF DIGITS = " +
reverseCount + "]");
        } else {
             System.out.println("Not
a REVSPY number [SUM OF DIGITS = " +
sum + ", REVERSE OF NUMBER OF DIGITS
= " + reverseCount + "]");
        }
    }

    // Method to reverse a number
  private static int reverseNumber(int
n) {
        int rev = 0;
        while (n > 0) {
            rev = rev * 10 + n % 10;
            n /= 10;
        }
        return rev;
    }
}
```

NODIA

```
        while (temp > 0) {
            int digit = temp % 10;
            product *= digit;
            count++;
            temp /= 10;
        }

         // Check if the product of
digits equals the number of digits
        if (product == count) {
      System.out.println("PERFECTSPY
number [PRODUCT OF THE DIGITS = " +
product + ", NUMBER OF DIGITS = " +
count + "]");
        } else {
             System.out.println("Not
a PERFECTSPY number [PRODUCT OF THE
DIGITS = " + product + " is not equal
to " + count + "]");
        }
    }
}
```

87. Define a class to accept a number and check whether it is a MAGICSPY number or not. A number is called MAGICSPY if the difference between the sum of its digits and the product of its digits equals the number of digits.
Example 1:
Input: 123
Output:
MAGICSPY number [SUM OF DIGITS = 1 + 2 + 3 = 6, PRODUCT OF DIGITS = 1 × 2 × 3 = 6, DIFFERENCE = 6 - 6 = 3, NUMBER OF DIGITS = 3]

Ans : MAIN 2013

```
import java.util.Scanner;

public class MagicSpyNumber {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Accept the input number
          System.out.print("Enter a
number: ");
        int num = sc.nextInt();

         // Initialize variables for
sum, product, and digit count
          int sum = 0, product = 1,
count = 0;
        int temp = num;

         // Calculate sum of digits,
product of digits, and count digits
        while (temp > 0) {
           int digit = temp % 10; //
Extract the last digit
           sum += digit;          //
Add to the sum
           product *= digit;      //
Multiply to the product
           count++;               //
Increment the digit count
           temp /= 10;            //
Remove the last digit
        }

          // Check if the difference
between sum and product equals the
number of digits
       if ((sum - product) == count)
{
        System.out.println("MAGICSPY
number [SUM OF DIGITS = " + sum + ",
PRODUCT OF DIGITS = " + product +
                                   ",
DIFFERENCE = " + (sum - product) + ",
NUMBER OF DIGITS = " + count + "]");
        } else {
             System.out.println("Not
a MAGICSPY number [SUM OF DIGITS = "
+ sum + ", PRODUCT OF DIGITS = " +
product +
                                   ",
DIFFERENCE = " + (sum - product) + "
≠ " + count + "]");
        }
    }
}
```

88. Define a class to accept a number and check whether it is a SQUARESPY number or not. A number is called SQUARESPY if the square of the sum of its digits equals the product of its digits.
Example 1:
Input: 112
Output:
SQUARESPY number [SUM OF DIGITS = 1 + 1 + 2 = 4, SQUARE OF SUM = 4^2 = 16, PRODUCT OF DIGITS = 1 × 1 × 2 = 16]

Ans : SQP 2012

```
import java.util.Scanner;

public class SquareSpyNumber {
    public static void main(String[]
args) {
```

Click Here to Buy book on Amazon

(i) What kind of variables are a,c and b declared in statement 1?
(ii) What kind of variable is declared and initialized in statement 2?
(iii) What result gets stored in a, b, and c within the method sum()?

Ans : MAIN 2011

(i) In statement 1, a, c, and b are member variables, also called instance variables.
(ii) In statement 2, a is a local variable.
(iii) a = 100, b = 10 (as c = a + b),
therefore, c = 110.
[Note : The local scope hides the global scope.]

79. Find the errors and correct them:

```
class Data {
    int result(int x, int y) {
        int z = x * y;
        return z;
    }

    double compute(int a, int b) {
        double c;
        c = a * b * z / a;
        return c;
    }

    void display() {
        System.out.println(z);
        d = compute(5, 10);
        System.out.println(c);
    }
}
```

Ans : SQP 2023

Error 1: c = a * b * z / a;
Reason: z is local to method result(). It cannot be accessed in compute().
Correction: c = a * b / a;
Error 2: System.out.println(c);
Reason: c is not accessible in display().
Correction: Add proper variable declaration or adjust scope.

80. Find the errors in the following program and correct them:

```
class Simple {
    int a, b, c;

    double sumNumber() {
        int d = a + b + c;
        double e = d / 5.6;
        return e;
    }

    void showNumber() {
        a = 10; b = 20; c = 5;
        System.out.println(d);
       double f = sumNumber(a, b, c);
        System.out.print(e);
    }
}
```

Ans : COMP 2024

Error 1: System.out.println(d);
Reason: d has a scope local to method sumNumber(). It cannot be accessed in showNumber().
Correction: Declare d properly in the class if required or remove the line.
Error 2: System.out.print(e);
Reason: e has a local scope in sumNumber().
Correction: Use a valid variable in showNumber() or print f.
Correction Code: System.out.print(f);

FIFTEEN MARKS QUESTIONS

81. Define a class to accept a string and convert it to uppercase. Create and display a new string by replacing each vowel with the previous character in the ASCII sequence and each consonant with the next character in the ASCII sequence. Any other characters (digits or symbols) should remain unchanged.
Example :
Input : #PROGRAMMING@2024
Output : #QSPQBNNJPH@2024

Ans : COMP 2020

```
import java.util.Scanner;

public class StringTransformation {
    public static void main(String[]
args) {
      Scanner sc = new Scanner(System.
in);

        // Accept the input string
          System.out.print("Enter a
string: ");
```

CHAPTER 7

ARRAYS

SUMMARY

1. ARRAY

An array is a finite collection of homogeneous data elements stored in contiguous memory locations.

1.1 Array Variable (Subscripted Variable)

An array variable, also known as a subscripted variable, is a user-defined name used to represent a set of adjacent memory locations.
For example:

```
int a[10];
```

Here, a is the array variable, and it reserves 10 memory locations for storing data elements.

1.2 Subscript or Index

A subscript or index is an integer value used to identify specific memory locations within an array.
For example:

```
int a[10];
```

In this case, the array size n is 10, so valid subscripts for the array a range from 0 to 9.

1.3 Types of Arrays

Java supports different categories of arrays based on their dimensions:

1. Single-dimensional arrays : Also referred to as one-dimensional arrays or single-subscripted variables.
2. Double-dimensional arrays : Arrays with two dimensions, often used for representing tables or matrices.
3. Multi-dimensional arrays : Arrays with more than two dimensions for handling complex data structures.

General format of Java statement for allocating memory locations to the single subscripted array variable:

```
datatype array_variable_name[] = new datatype[s];
```

Where s represents the number of memory locations reserved for the array.

2. UNDERSTANDING ARRAYS IN JAVA

2.1 Operator new

The new keyword in Java is used to allocate memory for arrays dynamically.

Example for allocating memory to one-dimensional array:

```
int weekdays[] = new int[7];
```

Diagrammatic representation of a one-dimensional array:

```
a[0]   a[1]   a[2]   a[3]   a[4]
a[5]   a[6]   a[7]   a[8]   a[9]
```

NODIA

3. CREATING AN ARRAY

In Java,the process of array creation involves two steps:

1. First step involves declaration of an array variable.
2. Second step involves allocation of memory to the array variable.

4. INITIALIZING A ONE-DIMENSIONAL ARRAY WITH GIVEN VALUES

General format:

```
data_type array_variable[] = {value1, value2, value3, ..., valueN};
```

Example:

```
double num[] = {2.5, 3.9, 1.2, 6.9};
```

5. DOUBLE DIMENSIONAL ARRAY

A two-dimensional array is represented using two subscripts:

1. The first subscript represents rows.
2. The second subscript represents columns.

It is sometimes referred to as an “array of arrays.”

Syntax:

```
datatype array_variable[][] = new datatype[r][c];
```

the size of an array. Its behavior differs slightly for one-dimensional and multi-dimensional arrays:

28. What does the following code do?

```
int x[ ][ ] = {{5, 2}, {-1, 0}, {30, 3}};
System.out.println(x.length);
```

(a) Prints the number of rows in the array.
(b) Prints the number of columns in the array.
(c) Prints the total number of elements in the array.
(d) Throws an error.

Ans : COMP 2012

(a) Prints the number of rows in the array.
In the given code:
int x[ ][ ] = {{5, 2}, {-1, 0}, {30, 3}};
System.out.println(x.length);

(i) Structure of x:
  (a) The array x is a 2D array with 3 rows and varying elements per row.
  (b) Each row is a one-dimensional array.
  (c) The .length property applied to a 2D array (e.g., x.length) returns the number of rows in the array.

(ii) Output of x.length : length evaluates to 3, as there are 3 rows in the array.

29. Which sorting technique repeatedly swaps adjacent elements if they are in the wrong order?
(a) Selection Sort
(b) Bubble Sort
(c) Merge Sort
(d) Quick Sort

Ans : MAIN 2016

(b) Bubble Sort
Bubble Sort is a simple sorting technique that works by repeatedly swapping adjacent elements if they are in the wrong order. The process is repeated until the array is sorted.

30. In selection sort, what happens during each iteration?
(a) Adjacent elements are compared and swapped if necessary.
(b) The largest element is moved to its correct position.
(c) The smallest element is found and swapped with the current element.
(d) All elements are divided into two halves and sorted recursively.

Ans : SQP 2020

(c) The smallest element is found and swapped with the current element.
Selection Sort is a simple sorting algorithm where the smallest (or largest, depending on the sorting order) element is repeatedly selected from the unsorted portion of the array and swapped with the first element of that portion. This process is repeated for all positions in the array until the entire array is sorted.

31. Which of the following describes the term "array of arrays"?
(a) A single-dimensional array.
(b) A two-dimensional array where each row is a separate array.
(c) A dynamically resized array.
(d) A special array that can hold different data types.

Ans : COMP 2004

(b) A two-dimensional array where each row is a separate array.
An array of arrays refers to a two-dimensional array in Java where each row is represented as a separate array. This structure allows each row to have a different number of elements, creating a jagged array.

32. What is the correct syntax to initialize a two-dimensional array in Java with specific values?
(a) int x[2][2] = {{1, 2}, {3, 4}};
(b) int x[ ][ ] = {{1, 2}, {3, 4}};
(c) int x[2][2] = [1, 2, 3, 4];
(d) int x = {{1, 2}, {3, 4}};

Ans : MAIN 2006

(b) int x[ ][ ] = {{1, 2}, {3, 4}};
To initialize a two-dimensional array in Java with specific values, you use a syntax where each row is enclosed in curly braces {} and the rows are enclosed in an outer pair of curly braces {}.

(i) This creates a 2x2 array and initializes it with the specified values.

(ii) The outer braces {} define the entire array, while the inner braces {1, 2} and {3, 4} define the rows.

33. What will happen if you try to access an array element with an index greater than its length?
(a) It will return null.
(b) It will return 0.
(c) It will throw an Array Index out of Bounds Exception.
(d) It will create a new element in the array.

Downloaded from www.pdf.tube

12. SEARCHING AN ITEM

Searching in an array means locating an item and returning its index or position.

Example: Searching for a contact in your phone.

Two commonly used techniques are:

(i) Linear Search

(ii) Binary Search

13. LINEAR SEARCH

(i) This type of search occurs in an unarranged array of values.

(ii) The value is searched starting from the first array element until it is found or until the end of the array.

(iii) It is a time-consuming method if the size of the array is large.

14. BINARY SEARCH

(i) This search occurs in an array arranged in ascending or descending order.

(ii) The middle subscript of the array is calculated.

(iii) The value to be searched is compared with the middle-positioned array element.

(iv) If the value is less than the middle element, the upper bound is adjusted, and the process continues in the first half of the array until the value is located.

(v) This process reduces the number of comparisons and is faster than linear search.

15. SORTING

The process of arranging an array in ascending or descending order is called sorting.

Two techniques of sorting in the syllabus:

(i) Bubble Sort

(ii) Selection Sort

15.1 Bubble Sorting

The adjacent elements of the array are compared.

If the array needs to be arranged in ascending order, and the left element (say a[0]) is greater than the right element (say a[1]), they need to be exchanged.

The process of exchanging values is commonly called swapping.

Download INDIA's Best Study app NODIA

15.2 Selection Sorting

To arrange an array in ascending order:

The smallest element is selected and exchanged with the first position element in the array.

The process is repeated for the remaining elements until all elements are arranged.

For descending order, the largest element is selected at each step.

16. PASSING AN ARRAY TO A METHOD

In Java, arrays are treated as objects. When an array is passed to a method:

(i) A reference to the array is passed, not a copy of its elements.

(ii) Any modifications made to the array within the method will directly affect the original array.

This behavior allows efficient handling of large arrays and avoids unnecessary duplication of data.

17. ARRAY OF OBJECTS

An array of objects is used when multiple instances of a class are required.

Each object is accessed using subscripts ranging from 0 to size - 1.

Example of an Array of Objects:

```
class Abc {
    int a;
    Abc() {
        a = 0;
    }
    void show() {
        a = a + 1;
         System.out.println("Value =
" + a);
    }
}
public   static   void   main(String
args[]) {
    Abc obj[] = new Abc[10];
    for (int i = 0; i < 10; i++) {
        obj[i].show();
    }
}
```

NODIA

Download INDIA's Best Study app NODIA

MULTIPLE CHOICE QUESTION

1. Which of the following statements is correct for initializing a String as "A robot is a programmable machine"?

(a) char x[ ] = {A, r, o, b, o, t, i, s, a, p, r, o, g, r, a, m, m, a, b, l, e, m, a, c, h, i, n, e};

(b) String x = "A robot is a programmable machine";

(c) StringBuffer x = new StringBuffer("A robot is a programmable machine");

(d) All of the above.

Ans : MAIN 2024

(d) All of the above

In Java, all the provided options are valid ways to represent the string "A robot is a programmable machine". Option (a) uses a char array, which stores individual characters. Option (b) initializes a String, the most common and immutable text type. Option (c) uses a StringBuffer, a mutable sequence of characters. Hence, all are correct.

2. For the following code, what is the output?

```
char[] myName = {A, D, T, T, I};
int someNumber = myName.length;
String someString = new String(myName);
int someOtherNumber = someString.length();
System.out.println(someNumber + "*" + someOtherNumber);
```

(a) Error

(b) 55

(c) 5

(d) 50

Ans : SQP 2023

(b) 55

(i) char[ ] myName : The array myName is initialized with characters {A, D, T, T, I}. This char[ ] has a length of 5.

(ii) int someNumber = myName.length : The length property of the char[] gives the number of elements, which is 5.

(iii) String someString = new String(myName) ; A String is created from the char[ ]. The String contains the characters "ADTTI", and its length is also 5.

(iv) int someOtherNumber = someString.length() : The length() method of the String returns the number of characters in the string, which is 5.

(v) System.out.println(someNumber + "*" + someOtherNumber);: The values of someNumber and someOtherNumber are concatenated with "*". This results in "55".

3. A single-dimensional array contains N elements. What will be the last subscript?

(a) N

(b) N-1

(c) N-2

(d) N+1

Ans : COMP 2022

(b) N-1

In a single-dimensional array in Java, the indexing starts at 0. If an array contains N elements, the first element has a subscript of 0, and the last element has a subscript of N-1.

For example:

```
int[] arr = {10, 20, 30, 40}; // Array with 4 elements
System.out.println(arr[3]);  // Last element is at index 4-1 = 3
```

4. What is the value of temperature[5]? if the list of temperatures is defined as below

int[ ] temperature = {30, 40, 70, 10, 55};

(a) 40

(b) 55

(c) ArrayIndexOutOfBoundsException

(d) 30

Ans : MAIN 2016

(c) ArrayIndexOutofBoundsException

The array temperature is defined with 5 elements: {30, 40, 70, 10, 55}. In Java, arrays are 0-indexed, meaning the valid indices for this array are 0 to 4.

(i) Attempting to access temperature[5] refers to the 6th element, which does not exist.

(ii) This results in an Array Index out of Bounds Exception, indicating an invalid index was accessed.

5. What is the pre-requisite for performing binary search in a one-dimensional array?

(a) No pre-requisite needed

(b) Array should be sorted

(c) Order of sorting should be known

(d) Both (b) and (c)

Ans : MAIN 2020

Click Here to Buy book on Amazon

(d) Both (b) and (c)

To perform binary search on a one-dimensional array, the following prerequisites must be met:

(i) Array should be sorted (b) : Binary search works by repeatedly dividing the array into halves and comparing the target with the middle element. This logic requires the array to be sorted.

(ii) Order of sorting should be known (c) : The algorithm needs to know whether the array is sorted in ascending or descending order to correctly navigate through the search.

6. A set of ice cubes in an ice-tray is an example of what type of array?

(a) One dimension

(b) Two dimensions

(c) Three dimensions

(d) Multi-dimension

Ans : SQP 2004

(b) Two dimensions

An ice tray typically represents a two-dimensional array, as it consists of rows and columns holding individual ice cubes. Each ice cube can be identified using two indices: one for the row and one for the column.

For example:

A tray with 3 rows and 4 columns can be visualized as a 2D array with dimensions 3×4.

7. What is the size of a Java array containing 4 characters?

(a) 4 bytes

(b) 4 bits

(c) 8 bytes

(d) 8 bits

Ans : SQP 2006

(c) 8 bytes

In Java, each char takes 2 bytes (16 bits) because Java uses Unicode encoding. For an array containing 4 char elements, the total memory required for the characters is : 4×2=8bytes

This does not include additional overhead used by the array object itself (like meta-data for length). Based on the given options, the correct answer focuses only on the characters' size.

8. Name the search or sort algorithm that, at each step, compares the sought key value with the key value of the middle element of the array.

(a) Linear search

(b) Binary search

(c) Insertion search

(d) None of the above

Ans : COMP 2010

(b) Binary search

Binary search is a search algorithm that, at each step :

(i) Compares the sought key value with the middle element of a sorted array.

(ii) If the key matches, the search ends.

(iii) Otherwise, the search continues in the left or right half, depending on whether the key is smaller or larger than the middle element.

This process is repeated until the key is found or the array is exhausted.

Linear search (a) scans each element sequentially, and insertion search (c) is not a valid algorithm in this context.

9. What is the output of the below code segment?

```
int x[ ] = {4, 3, 7, 8, 9, 10};
System.out.print(x.length + " ");
System.out.print(x[1] + x[5] * x[3]);
```

(a) 6 83

(b) 5 67

(c) 6 67

(d) 5 83

Ans : MAIN 2013

(a) 6 83

(i) x.length:

The x array has 6 elements: {4, 3, 7, 8, 9, 10}. The length property returns the number of elements in the array, which is 6.

(ii) Expression x[1] + x[5] * x[3]:

x[1] = 3 (second element of the array)

x[5] = 10 (sixth element of the array)

x[3] = 8 (fourth element of the array)

Substituting these values:

= x[1] + x[5] * x[3]

= 3 + (10 * 8) = 3 + 80 = 83

(ii) Final Output:

The first System.out.print prints 6, and the second prints 83.

NODIA

Download INDIA's Best Study app NODIA

r = No. of rows reserved for array.
c = No. of columns reserved for array.

Example:

```
double performance[][] = new double[10][4];
```

6. DECLARING TWO-DIMENSIONAL ARRAYS

```
datatype array_variable[][] = new datatype[r][c];
```

r: Number of rows reserved for the array.
c: Number of columns reserved for the array.

Example:

```
double performance[][] = new double[10][4];
```

This declaration creates a 2D array named performance with 10 rows and 4 columns.

7. DIAGRAMMATIC REPRESENTATION OF A 2-D ARRAY

Suppose an array
A[3][4] = {{-30, 2, 100, 66, 55}, {11, -1, -33, 45, 49}, {69, 89, 98, -95, 99}, {29, 19, 61, -33, 87}}.
It can be represented diagrammatically as follows:

```
Columns →   0     1     2     3     4
Rows ↓
0         -30     2   100    66    55
1          11    -1   -33    45    49
2          69    89    98   -95    99
3          29    19    61   -33    87
```

```
Here
A[0][0] = -30,   A[0][1] = 2     and so on
A[1][0] = 11,    A[1][1] = -1    and so on
A[2][0] = 69,    A[2][1] = 89    and so on
A[3][0] = 29,    A[3][1] = 19    and so on
```

8. INITIALIZING A 2-D ARRAY

Example:

```
int x[][] = {{5, 2}, {-1, 0}, {30, 3}};
```

The above array x is initialized for 3 rows and 2 columns.

9. LENGTH OF AN ARRAY

The length of an array in Java represents the number of elements it contains. This property varies for one-dimensional and two-dimensional arrays.

Download INDIA's Best Study app NODIA

9.1 Length of One-Dimensional Array

It retrieves the allocated/reserved locations for the array.
Syntax:

```
int variable = arrayvariable.length;
```

Example:

```
int days[] = {10, 30, 100, 95, 59};
int y = days.length;
System.out.print(y);
```

Output:

```
5
```

9.2 Length of Two-Dimensional Array

Syntax:

```
int variable = arrayvariable[r].length;
```

Here, r represents the row whose length will be returned.

Example:

```
int x[][] = new int[3][50];
System.out.println(x[0].length);
System.out.println(x[1].length);
System.out.println(x[2].length);
```

Output:

```
50
50
50
```

Explanation

The array x is an array of three arrays. Each of these 3 arrays is of the same length 50 (3 rows, 50 columns).

10. ARRAYS: A CONCEPTUAL AND PRACTICAL OVERVIEW

Have You Seen Arrays?

1. An egg tray
2. A set of water bottles
3. A chessboard or calendar

11. USES OF ARRAYS

Arrays are essential for organizing and manipulating data efficiently.

1. Counting Items : Simplifies tracking and managing items systematically.
2. Mathematical Operations : Supports calculations like addition, multiplication, and division.
3. Searching : Enables quick location of specific items in a dataset.
4. Sorting : Facilitates arranging elements in ascending, descending, or alphabetical order.
5. Data Storage : Handles large datasets efficiently for dynamic processing.

Downloaded from www.pdf.tube

15. In the declaration int x[ ][ ] = new int[3][50];, what does x[0].length return?

(a) 3

(b) 50

(c) 150

(d) Undefined

Ans : MAIN 2022

(b) 50

In the declaration int x[ ][ ] = new int[3][50];, a 2D array is created with 3 rows and 50 columns. The .length property gives the size of the corresponding dimension:

x.length: Returns 3 because there are 3 rows.

x[0].length: Refers to the length of the first row (x[0]), which has 50 elements (columns).

16. Which of the following search techniques works on unsorted arrays?

(a) Linear Search

(b) Binary Search

(c) Bubble Search

(d) Selection Search

Ans : SQP 2015

(a) Linear Search

Linear search is the only search technique that works on unsorted arrays. It sequentially checks each element of the array until the desired element is found or the end of the array is reached.

Why Other Options Are Incorrect:

(i) (b) Binary Search : Requires the array to be sorted for efficient searching.

(ii) (c) Bubble Search : This is not a valid search algorithm; "Bubble" is associated with sorting (Bubble Sort).

(iii) (d) Selection Search : This is also not a valid algorithm; "Selection" is related to sorting (Selection Sort).

17. In binary search, what happens when the value to be searched is less than the middle element?

(a) The search continues in the second half of the array.

(b) The search stops, as the value cannot be found.

(c) The search continues in the first half of the array.

(d) The array is rearranged.

Ans : COMP 2018

(c) The search continues in the first half of the array.

Binary search works by repeatedly dividing a sorted array into two halves and narrowing the search range. The steps are as follows:

(i) Calculate the middle element: The middle index is computed as (low + high)

(ii) Compare the value to be searched with the middle element:

(a) If the value is equal to the middle element, the search ends successfully.

(b) If the value is less than the middle element, the search continues in the first half of the array (lower indices).

(c) If the value is greater than the middle element, the search continues in the second half of the array (higher indices).

NODIA

18. In bubble sorting, what is the purpose of swapping?

(a) To move larger elements to the end of the array.

(b) To sort elements in descending order only.

(c) To add new elements to the array.

(d) To remove duplicate elements from the array.

Ans : MAIN 2019

(a) To move larger elements to the end of the array.

Bubble sort works by repeatedly comparing adjacent elements in an array and swapping them if they are in the wrong order. The goal of swapping is to gradually move the larger elements toward the end of the array (or the smaller elements to the beginning, depending on the sort order).

19. What happens when an array is passed to a method in Java?

(a) A copy of the array elements is passed.

(b) A reference to the array is passed.

(c) Only the first element of the array is passed.

(d) The array cannot be passed to a method.

Ans : SQP 2009

(b) A reference to the array is passed.

In Java, when an array is passed to a method, a reference to the original array is passed, not a copy of the array or its elements. This means any changes made to the array inside the method will affect the original array.

20. How is an array of objects created in Java?

(a) Abc obj = new Abc[10];

(b) Abc obj[ ] = new Abc[10];

(c) Abc[ ] obj = Abc[10];

(d) obj[ ] = new Abc[10];

10. What is an array?
(a) A collection of heterogeneous data elements stored in separate memory locations.
(b) A finite collection of homogeneous data elements stored in contiguous memory locations.
(c) A collection of elements stored in random memory locations.
(d) A collection of data elements with different data types.

Ans : SQP 2014

(b) A finite collection of homogeneous data elements stored in contiguous memory locations.

An array in Java is :
(i) A finite collection, meaning it has a fixed size defined at the time of creation.
(ii) A collection of homogeneous data elements, meaning all elements must be of the same data type.
(iii) Stored in contiguous memory locations, enabling efficient indexing and traversal.

Other options :
(i) (a) and (d) describe heterogeneous collections, not arrays.
(ii) (c) suggests random memory locations, which is incorrect for arrays.

11. In the declaration int a[10];, what does a represent?
(a) An integer variable
(b) A single element of the array
(c) An array variable that reserves 10 memory locations
(d) A constant value

Ans : COMP 2011

(c) An array variable that reserves 10 memory locations
In the declaration int a[10];, the following holds true :
(i) a is an array variable that refers to a block of memory capable of storing 10 integers.
(ii) The declaration allocates contiguous memory locations for storing the 10 integer elements, indexed from 0 to 9.

12. Which of the following correctly initializes a one-dimensional array?
(a) double num[] = {2.5, 3.9, 1.2, 6.9};
(b) int num[] = new int[] {2.5, 3.9, 1.2, 6.9};
(c) int num[4] = (2.5, 3.9, 1.2, 6.9);
(d) float num[] = new {2.5, 3.9, 1.2, 6.9};

Ans : MAIN 2007

(a) double num[] = {2.5, 3.9, 1.2, 6.9};
(i) Option (a) : This is the correct syntax for initializing a one-dimensional array in Java. The array is declared as a double array, and the values {2.5, 3.9, 1.2, 6.9} are provided directly during initialization.
(ii) Option (b) : Incorrect because the data type is int, but the values provided are floating-point numbers, which cannot be stored in an int array without explicit casting.
(iii) Option (c) : Incorrect syntax. Java does not use ( ) for array initialization, and specifying the array size ([4]) during initialization with specific values is not allowed.
(iv) Option (d) : Incorrect syntax. The new keyword must be followed by the data type and cannot omit it as shown here.

13. What does .length return when used with a one-dimensional array?
(a) The total memory allocated for the array
(b) The number of elements in the array
(c) The index of the last element in the array
(d) The data type of the array elements

Ans : SQP 2008

(b) The number of elements in the array
The length property of a one-dimensional array in Java returns the number of elements in the array, not the memory allocated or indices.
(i) length does not provide memory details.
(ii) The index of the last element is array.length - 1.
(iii) length does not indicate the data type.

14. What is the output of the following code?
```
int days[ ] = {10, 30, 100, 95, 59};
System.out.print(days.length);
```
(a) 4
(b) 5
(c) 6
(d) 10

Ans : COMP 2010

(b) 5
The days array is declared and initialized with 5 elements: {10, 30, 100, 95, 59}. The .length property of an array in Java returns the number of elements in the array.

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

The new operator creates an array and initializes it with default values (e.g., 0 for integers, null for objects).

24. What is the range of valid subscripts for an array declared as int a[10];?

(a) 1 to 10

(b) 0 to 10

(c) 0 to 9

(d) 1 to 9

Ans : MAIN 2017

(c) 0 to 9

In Java, arrays are 0-indexed, meaning the indexing starts from 0. For an array declared as int a[10];, the number of elements is 10, and the valid subscripts (indices) are:

0, 1, 2, 3, ..., 9

0, 1, 2, 3, ..., 9

The first element is accessed using a[0].

The last element is accessed using a[9]..

25. Which of the following is a correct representation of a two-dimensional array declaration?

(a) data type array_variable = new data type[rows][columns];

(b) data type array_variable[rows][columns];

(c) data type array_variable[ ][ ] = new data type[rows][columns];

(d) data type[ ][ ] array_variable = new data type[columns][rows];

Ans : COMP 2021

(c) data type array_variable[][] = new data type[rows][columns];

In Java, a two-dimensional array is declared and initialized using the following syntax:

java

```
data type array_variable[][] = new data type[rows][columns];
```

Key Points:

Two sets of brackets [ ][ ] : Represent the two dimensions of the array.

new datatype [rows][columns] : Dynamically allocates memory for the array with the specified number of rows and columns.

Declaration and initialization : You can declare the array and allocate memory in the same statement or separately.

Example:

java

```
int[][] matrix = new int[3][4]; // Creates a 3x4 two-dimensional array.
```

26. What is the key difference between linear search and binary search?

(a) Linear search is faster than binary search.

(b) Binary search works only on sorted arrays, while linear search works on both sorted and unsorted arrays.

(c) Linear search divides the array into two halves, while binary search scans each element.

(d) Binary search requires more time than linear search.

Ans : MAIN 2024

(b) Binary search works only on sorted arrays, while linear search works on both sorted and unsorted arrays.

The key difference between linear search and binary search lies in their application and efficiency:

(i) Linear Search:
- (a) It works on both sorted and unsorted arrays.
- (b) It sequentially checks each element until the desired element is found or the array ends.
- (c) Time complexity : O(n), where n is the number of elements.

(ii) Binary Search:
- (a) It only works on sorted arrays.
- (b) It divides the array into two halves and repeatedly narrows the search range by comparing the target value to the middle element.
- (c) Time complexity : (log)O(logn), making it significantly faster for large arrays compared to linear search.

27. Which of the following is true about the .length property in Java?

(a) It can only be used with single-dimensional arrays.

(b) It returns the total memory size of the array.

(c) It returns the number of elements in a one-dimensional array or the size of a row in a two-dimensional array.

(d) It always returns the size of the array minus one.

Ans : SQP 2023

(c) It returns the number of elements in a one-dimensional array or the size of a row in a two-dimensional array.

The .length property in Java is used to determine

Downloaded from www.pdf.tube

Ans : COMP 2003

(b) Abc obj[] = new Abc[10];
In Java, an array of objects is created using the following syntax:

(i) Class Name[] declares an array of references to objects of the class Class Name.

(ii) new Class Name[size] allocates memory for the array and initializes it with null references.

(iii) You need to explicitly initialize each element of the array with an instance of the class before using it.

21. In the following example, what does obj[i].show(); do?

```
class Abc {
    int a;
    Abc() {
        a = 0;
    }
    void show() {
        a = a + 1;
         System.out.println("Value =
" + a);
    }
}
public   static   void   main(String
args[]) {
    Abc obj[] = new Abc[10];
    for (int i = 0; i < 10; i++) {
        obj[i] = new Abc();
        obj[i].show();
    }
}
```

(a) Displays the value of a after incrementing it by 1.

(b) Creates a new array of objects.

(c) Initializes a to 0 for each object.

(d) Prints the index of each object in the array.

Ans : MAIN 2002

(a) Displays the value of a after incrementing it by 1.
Yes, the obj[i].show(); statement displays the value of a after incrementing it by 1.

(i) Initialization : Each object in the array obj[i] is initialized using the default constructor of the Abc class, which sets a = 0.

(ii) Increment in show() Method : The show() method increments a by 1 with a = a + 1 and then prints it.

(iii) Display : The method prints the updated value of a.

For each iteration in the loop, the output will be: Value = 1
Since a starts at 0 for every object and is incremented to 1 inside show(), the displayed value will always be 1 for each call to obj[i]. show();.

22. What is the general syntax for declaring a single-dimensional array in Java?

(a) datatype array_variable = new datatype[size];

(b) datatype array_variable[] = new datatype[size];

(c) datatype[] array_variable = new size;

(d) array_variable datatype[] = new size[datatype];

Ans : COMP 2001

(b) datatype array_variable[] = new datatype [size];
In Java, the general syntax for declaring and initializing a single-dimensional array is:
datatype array_variable[ ] = new datatype[size];
Explanation of the components:

(i) datatype : Specifies the type of elements the array will hold (e.g., int, float, String, etc.).

(ii) array_variable[ ] : Declares a variable to reference the array.

(iii) new datatype[size] : Allocates memory for the array and specifies the number of elements it can hold.

int numbers[ ] = new int[5]; // Declares an array of integers with size 5.

23. In Java, which operator is used to dynamically allocate memory for arrays?

(a) Malloc

(b) Allocate

(c) New

(d) Init

Ans : SQP 2005

(c) New
In Java, the new operator is used to dynamically allocate memory for arrays and objects. When an array is created using new, memory is allocated on the heap, and all elements of the array are initialized to their default values based on the data type.
Example :
java
Copy code
int[] numbers = new int[5]; // Allocates memory for 5 integers.

NODIA

Ans : SQP 2010

(c) It will throw an Array Index Out Of Bounds Exception.
In Java, if you try to access an array element using an index greater than or equal to the array's length, the Java Runtime Environment (JRE) throws an Array Index out of Bounds Exception. This exception occurs because the array does not have memory allocated for indices beyond its declared size.

34. Which search technique is faster for large, sorted arrays?

(a) Linear Search
(b) Bubble Search
(c) Binary Search
(d) Sequential Search

Ans : COMP 2013

(c) Binary Search
For large, sorted arrays, Binary Search is significantly faster than Linear Search or any other simple searching technique because it reduces the search space by half at each step.
Why Binary Search is Faster:

(i) Binary Search works by repeatedly dividing the array into two halves:
  (a) If the target value is equal to the middle element, the search ends.
  (b) If the target value is smaller than the middle element, the search continues in the left half.
  (c) If the target value is greater, the search continues in the right half.

(ii) This divide-and-conquer approach results in a time complexity of O(logn), which is much faster than the O(n) time complexity of Linear Search for large arrays.

35. What does the following code do?
int arr[ ] = {10, 20, 30, 40};
arr[1] = 25;
System.out.println(arr[1]);

(a) Prints 20
(b) Prints 25
(c) Throws an error because arrays are immutable
(d) Prints 30

Ans : MAIN 2014

(b) Prints 25
In Java, arrays are mutable, meaning their elements can be modified after the array is created. In the given code:
int arr[ ] = {10, 20, 30, 40};
arr[1] = 25;
System.out.println(arr[1]);

(i) The array arr is initialized with the values {10, 20, 30, 40}.
(ii) The statement arr[1] = 25; updates the second element of the array (index 1) to 25.
(iii) The System.out.println(arr[1]); prints the updated value, which is 25.

36. Which statement about arrays in Java is correct?

(a) Arrays in Java are immutable.
(b) Arrays can store different types of data in the same array.
(c) Arrays are fixed in size and homogeneous.
(d) Arrays do not require memory allocation.

Ans : SQP 2011

(c) Arrays are fixed in size and homogeneous.
In Java, arrays are a fundamental data structure with the following properties:

(i) Fixed Size: The size of an array is determined at the time of its creation and cannot be changed. For example: int[] arr = new int[5]; // Creates an array with 5 elements.

(ii) Homogeneous Data: Arrays in Java are homogeneous, meaning all elements in an array must be of the same data type (e.g., int, String, double).

(iii) Memory Allocation: Arrays require explicit memory allocation using the new keyword. For example: int[] arr = new int[5]; // Allocates memory for 5 integers.

(iv) Mutable Nature: While the size of an array is fixed, the contents (elements) of an array can be modified.

37. What will be the result of the following code?

```
java
int x[ ][ ] = new int[3][4];
System.out.println(x.length);
```

(a) 3
(b) 4
(c) 12
(d) Throws an error

Ans : COMP 2007

(a) 3
It refers to the number of rows.
In the given code:

Click Here to Buy book on Amazon

java Copy code int x[][] = new int[3][4]; System.out.println(x.length);

(i) Understanding x.length:
  (a) In a 2D array, the .length property on the array variable (e.g., x.length) refers to the number of rows in the array.
  (b) Here, x has 3 rows, so x.length returns 3.

(ii) Structure of x:
  (a) The array x is a 2D array with 3 rows and 4 columns.
  (b) Each row is itself an array of size 4.
  (c) The total number of elements in the array is 3 * 4 = 12, but this is not what .length returns.

38. What is the output for the below statements?
int x[ ] = {1, 2, 3};
int y[ ] = {1, 2, 3};
if (x == y)
System.out.print("We are equal");
else
System.out.print("We are not equal");
(a) We are equal
(b) We are not equal
(c) "We are equal"
(d) "We are not equal"

Ans : MAIN 2008

(b) We are not equal
Arrays are compared by reference, not content.
Explanation:
In Java, when comparing arrays using the == operator, the reference (memory address) of the arrays is compared, not their contents. Even if two arrays have identical elements, they are stored in different memory locations and thus are not considered equal when compared using ==.

39. Which of the below operators is used to allocate memory to an array in Java programming language?
(a) new
(b) New
(c) Allocate
(d) Dot operator

Ans : SQP 2010

(a) new
In Java, the new operator is used to allocate memory dynamically for arrays and objects. When you create an array, new allocates the required amount of memory in the heap to store the array elements.

40. A developer writes the following code to assign a value to the first element of an array arr:
`arr[1] =+ 10;`
This is incorrect. What will be the correct statement?
A. arr[0] = 10;
B. arr[0] += 10;
C. arr[0] = arr[0] + 10;
D. Both B and C
(a) Only A
(b) Both A and B
(c) Both A and C
(d) Both B and C

Ans : COMP 2015

(d) Both B and C
The provided code arr[1] =+ 10; is incorrect because =+ is not a valid operator in Java. This assignment statement would be interpreted as arr[1] = (+10);, which simply assigns 10 to arr[1]. However, the intent of the question seems to focus on correcting the logic for updating the first element of the array (arr[0]), not the second element (arr[1]).

41. A programmer writes the following code to declare an array in Java:
`int arr[5];`
This is incorrect. What will be the correct statement?
A int arr[] = new int[5];
B int[] arr = new int[5];
C int arr[] = {1, 2, 3, 4, 5};
D All of the above
(a) Only A
(b) Only B
(c) Both A and C
(d) All of the above

Ans : MAIN 2018

(d) All of the above
The statement int arr[5]; is incorrect in Java because it doesn't allocate memory for the array. The correct way to declare an array includes allocating memory using new or initializing it directly. Both int arr[] = new int[5]; and int[] arr = new int[5]; are correct for memory allocation, and int arr[] = {1, 2, 3, 4, 5}; is correct for initialization.

NODIA
Download INDIA's Best Study app NODIA

42. A student writes the following code to calculate the length of an array nums:

```
int length = nums.size;
```

This is incorrect. What will be the correct statement?

A. int length = nums.length;
B. int length = nums[0].length;
C. int size = nums.length();
D. Both A and B

(a) Only A
(b) Only B
(c) Only C
(d) Both A and B

Ans : SQP 2019

(d) Both A and B
In Java, the length of an array is accessed using the length property, not a method like size() or length(). For a one-dimensional array, nums. length gives the total number of elements. For a two-dimensional array, nums[0].length provides the number of columns in the first row.

43. A programmer writes the following code to search for a value in an array using binary search:

```
if (value > mid) {
    search = "First Half";
}
```

This is incorrect. What will be the correct statement?

A. if (value < mid) { search = "First Half"; }
B. if (value > mid) { search = "Second Half"; }
C. if (value == mid) { search = "Found"; }
D. All of the above

(a) Only A
(b) Both A and B
(c) Both B and C
(d) All of the above

Ans : COMP 2024

(d) All of the above
In binary search, if the value is less than mid, it lies in the first half; if greater, it lies in the second half. If value equals mid, it is found. The correct statements cover all these conditions: A handles the first half, B handles the second half, and C confirms the match.

44. A student writes the following code to sort an array using bubble sort:

```
if (arr[i] < arr[i + 1]) {
    swap(arr[i], arr[i + 1]);
}
```

This is incorrect for ascending order. What will be the correct statement?

A. if (arr[i] > arr[i + 1]) { swap(arr[i], arr[i + 1]); }
B. if (arr[i] == arr[i + 1]) { continue; }
C. if (arr[i + 1] < arr[i]) { swap(arr[i + 1], arr[i]); }

(a) Only A
(b) Only C
(c) Both A and C
(d) Both B and C

Ans : MAIN 2023

(c) Both A and C
For bubble sort in ascending order, elements must be swapped when arr[i] > arr[i + 1]. The correct condition is A or its equivalent, C (if (arr[i + 1] < arr[i])). Option B is unnecessary for sorting. Hence, both A and C are correct.

45. Consider the following program segment where statements are jumbled. Choose the correct order of statements to initialize a one-dimensional array.

```
int arr[] = new int[5]; // (1)
arr[0] = 10;            // (2)
arr[1] = 20;            // (3)
arr[2] = 30;            // (4)
arr[3] = 40;            // (5)
arr[4] = 50;            // (6)
```

(a) (1) (2) (3) (4) (5) (6)
(b) (2) (3) (4) (1) (5) (6)
(c) (1) (6) (5) (4) (3) (2)
(d) (4) (1) (2) (3) (5) (6)

Ans : SQP 2012

(a) (1) (2) (3) (4) (5) (6)
To initialize a one-dimensional array correctly, the array must first be declared and memory allocated using int arr[] = new int[5];. After that, values can be assigned to the array elements in sequential order.

46. Consider the following program segment where statements are jumbled. Choose the correct order of statements to search for an element using linear search.

```
for (int i = 0; i < n; i++) {     //
(1)
if (arr[i] == key) {           // (2)
System.out.println("Found");       //
(3)
break;                         // (4)
}
}
```

Downloaded from www.pdf.tube

(a) (1) (3) (4) (2)
(b) (1) (2) (3) (4)
(c) (2) (1) (4) (3)
(d) (4) (1) (3) (2)

Ans : COMP 2016

(b) (1) (2) (3) (4)
To implement linear search, you must iterate through the array using a loop (1) and check if the current element matches the key (2). If a match is found, print "Found" (3) and exit the loop using break (4)

47. Consider the following program segment where statements are jumbled. Choose the correct order of statements to calculate the length of an array.

```
int arr[] = {10, 20, 30, 40}; // (1)
int length = arr.length;      // (2)
System.out.println(length);   // (3)
```

(a) (1) (2) (3)
(b) (2) (1) (3)
(c) (1) (3) (2)
(d) (3) (2) (1)

Ans : MAIN 2020

(a) (1) (2) (3)
To calculate and display the length of an array, the array must first be declared and initialized (1). Then, the length property can be accessed to determine the array's size (2), and finally, the length is printed (3).

48. Consider the following program segment where statements are jumbled. Choose the correct order of statements to implement bubble sort for an array.

```
if (arr[j] > arr[j + 1]) {  // (1)
temp = arr[j];              // (2)
arr[j] = arr[j + 1];        // (3)
arr[j + 1] = temp;          // (4)
}
```

(a) (1) (2) (3) (4)
(b) (4) (3) (2) (1)
(c) (2) (1) (4) (3)
(d) (1) (3) (4) (2)

Ans : SQP 2004

(a) (1) (2) (3) (4)
In bubble sort, we first compare adjacent elements (1). If the element on the left is greater than the one on the right, we perform a swap using a temporary variable. The swap steps are assigning arr[j] to temp (2), assigning arr[j + 1] to arr[j] (3), and assigning temp to arr[j + 1] (4).

NODIA

49. **Assertion (A) :** Arrays in Java are treated as objects.
Reason (R) : When an array is passed to a method in Java, the reference to the array is passed, and changes made within the method affect the original array.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2006

(a) Both A and R are true, and R is the correct explanation of A.
In Java, arrays are treated as objects, which is why Assertion (A) is true. When an array is passed to a method, the reference to the array is passed, meaning changes made to the array within the method affect the original array, making Reason (R) true as well. Furthermore, R correctly explains A.

50. **Assertion (A) :** The .length property can be used to determine the size of an array in Java.
Reason (R) : The .length property returns the total memory allocated for the array in Java.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2010

(c) A is true, but R is false.
The length property in Java is used to determine the number of elements in an array, not the total memory allocated for it. Thus, Assertion (A) is true, but Reason (R) is false because it incorrectly explains the purpose of the .length property.

51. **Assertion (A) :** Bubble sort is used to arrange elements of an array in ascending or descending order.
Reason (R) : In bubble sort, adjacent elements of the array are compared, and if they are not in the desired order, they are swapped.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : SQP 2013

(a) Both A and R are true, and R is the correct explanation of A.
Bubble sort is indeed used to arrange elements in ascending or descending order (A is true). The process involves comparing adjacent elements and swapping them if they are not in the desired order, which correctly explains how bubble sort works (R is true and explains A).

52. **Assertion (A) :** Binary search can be used on both sorted and unsorted arrays.
Reason (R) : Binary search divides the array into two halves and reduces the search space iteratively.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is false, but R is true.

(d) Both A and R are false.

Ans : COMP 2014

(c) A is false, but R is true.
Binary search requires the array to be sorted for it to function correctly, so Assertion (A) is false. However, Reason (R) correctly explains how binary search operates, by dividing the array into halves and reducing the search space iteratively.

53. **Assertion (A) :** A two-dimensional array is also known as an "array of arrays."
Reason (R) : In a two-dimensional array, elements are stored as a matrix where rows and columns are independent arrays.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is true, but R is false.

(d) A is false, but R is true.

Ans : MAIN 2011

(c) A is true, but R is false
A two-dimensional array in Java is indeed called an "array of arrays," as each row is essentially an independent array (A is true). However, while elements can be conceptually visualized as a matrix, rows and columns are not truly independent arrays. The structure depends on nested arrays, so R is false.

54. **Assertion (A) :** Linear search is a faster searching technique than binary search for large arrays.
Reason (R) : Linear search checks every element in the array until the desired element is found or the array ends, while binary search divides the array into two halves.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is false, but R is true.

(d) Both A and R are false.

Ans : SQP 2007

(c) A is false, but R is true.
Linear search is not faster than binary search for large arrays because it checks every element sequentially, resulting in a time complexity of O(n), while binary search operates with O(logn) complexity, making it much faster for sorted arrays. Thus, Assertion (A) is false. However, Reason (R) correctly describes how linear and binary search work.

55. **Assertion (A) :** In selection sort, the largest element is placed at the beginning of the array in ascending order.
Reason (R) : Selection sort selects the smallest element from the unsorted part of the array and swaps it with the first unsorted element.

(a) Both A and R are true, and R is the correct explanation of A.

(b) Both A and R are true, but R is not the correct explanation of A.

(c) A is false, but R is true.

(d) Both A and R are false.

Ans : COMP 2008

(c) A is false, but R is true.
In selection sort, for ascending order, the smallest element is placed at the beginning of the array, not the largest, so Assertion (A) is false. However, Reason (R) is true because selection sort works by selecting the smallest element from the unsorted portion and swapping it with the first unsorted element.

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

2.0
3.0
4.0

73. Write the output of the following program segments:

(i)

```
int a[] = {20, 30, 40, 50};
for (int i = 0; i < 4; i++) {
if (a[i] % 4 == 0) {
System.out.println(a[i]);
  }
}
```

(ii)

```
int h[] = {6, 12, 18, 24, 30, 42};
int sum = 0;
for (int k = 0; k <= 4; k++) {
sum += h[k];
}
System.out.print(sum);
```

Ans : COMP 2019

(i) 20
40
(ii) 90

74. If int x[] = {4, 3, 7, 8, 9, 10}; what are the values of p and q?
(i) p = x.length
(ii) q = x[2] + x[5] * x[1]

Ans : MAIN 2016

(i) p = 6 (length of array x[])
(ii) q = x[2] + x[5] * x[1] = 7 + 10 * 3 = 37

75. (i) What will this code print? *(Choose the correct option)*

```
int arr[] = new int[5];
System.out.println(arr);
```

(a) 0
(b) Value stored in arr[0]
(c) 0000
(d) Garbage value

(ii) Name the keyword which is used to resolve the conflict between method parameters and instance variables/fields.

Ans : COMP 2012

(i) Garbage value
(i) this keyword

76. Write Java statements for the following:
(i) Initialize the first element of a one-dimensional array A as 0.
(ii) Assign 65, 66, 67 to an array arr.
(iii) Check if an element at column 5 of an array x[10] is divisible by 2.
(iv) Initialize true, false, true, true to a boolean array.

Ans : MAIN 2016

(i)

```
a[0] = 0;
```

(ii)

```
int arr[] = {65, 66, 67};
```

(iii)

```
if (x[4] % 2 == 0)
System.out.print("Divisible");
else
System.out.print("NotDivisible");
```

(iv)

```
boolean a[] = {true, false, true,
true};
```

77. Consider the following Java statements and answer the questions that follow:

```
String a[] = {"FLORIDA", "AMAZON",
"WASHINGTON"};
```

(i) What is the range of indices for array a?
(ii) What is the result of a[2].length?
(iii) What is the size of the array?

Ans : SQP 2020

(i) 0 to 2
(ii) 10
(iii) 3 locations of String type.

78. What is the difference between int[ ] and int[10]?

Ans : COMP 2004

int[ ]: Declares a one-dimensional array of int type.
int[ ][ ]: Declares a two-dimensional array of int type.

79. Consider the following list of numbers:
5, 0, 2, 8, 1, 13, 9
Show the numbers list after the third iteration of Bubble Sorting in descending order.

Ans : MAIN 2006

After the third iteration:
8, 5, 13, 9, 2, 1, 0

NODIA

Ans : SQP 2010

```
int var[] = {4, 8, 12, 16, 20, 24};
```

69. Write a Java statement for the following purposes:
(i) Declare and create an array NUM of 40 integers.
(ii) Declare and create an array STR of 20 names.
(iii) Initialize an array of integers to store numbers -10, -8, -6, ..., 12.
(iv) Initialize an array of 5 names of convent schools:
ST. MARY, ST. THOMAS, ST. JOSEPHS ACADEMY, ST. PAULS ACADEMY, and ST. PETERS ACADEMY.
(v) Create an array of 10 real numbers named as REAL.

Ans : COMP 2022

(i)
```
int NUM[] = new int[40];
```
(ii)
```
String STR[] = new String[20];
```
(iii)
```
int nums[] = {-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10, 12};
```
(iv)
```
String schools[] = {"ST. MARY", "ST. THOMAS", "ST. JOSEPHS ACADEMY", "ST. PAULS ACADEMY", "ST. PETERS ACADEMY"};
```
(v)
```
float REAL[] = new float[10];
```

70. Name the search or sort algorithm that:
(i) Makes several passes through the array, selecting the next smallest item each time and placing it where it belongs.
(ii) At each state, compares the sought key value with the middle element of the array.

Ans : COMP 2007

(i) Selection Sort.
(ii) Binary Sort

71. Determine errors in the following statements of Java programs and correct them:
(i) int a[10] = {5, 2, 3, 4, 6, 7};
(ii) String x[ ] = {"JAVA", "C++"};
(iii)
```
int x[5] = {6, -2, 2, 3, 5, 6};
for (int i = 0; i <= 5; i++) {
  System.out.println(x[i]);
  }
```
(iv) char clr[ ] = {4, 6, 9, 2, 0};

Ans : MAIN 2015

(i) Correction : `int a[] = {5, 2, 3, 4, 6, 7};`

(ii) Correction : Correct as it is. No changes required.

(iii) Correction : The array x[5] declaration should match the elements or limit the loop to avoid out-of-bounds.
```
int x[] = {6, -2, 2, 3, 5, 6};
for (int i = 0; i < 5; i++) {
  System.out.println(x[i]);
  }
```

(iv) Correction : Correct as it is. No changes required.

72. Write the output of the following program:

(i)
```
int x[] = {-5, -4, -3, -2, 1};
for (int i = 3; i < 5; i++) {
   System.out.print(x[i] + " ");
  }
```

(ii)
```
String n[3] = {"OBJECT", "ORIENTED", "PROGRAMS"};
for (int y = 0; y < 3; y++) {
System.out.print(n[y] + " ");
}
```

(iii)
```
double x[] = {2.93, 4.65, 3.12, 9.0, 6};
for (int s = 0; s <= 4; s+1) {
x[s] += 5;
System.out.println(x[s]);
}
```

(iv)
```
   double x[] = new double[5];
for (int j = 0; j < 5; j++) {
    x[j] = j;
    System.out.println(x[j]);
}
```

Ans : SQP 2018

(i) -2 1
(ii) OBJECT ORIENTED PROGRAMS
(iii)
```
7.93
9.65
8.12
14.0
11.0
```
(iv) 0.0
1.0

NODIA

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

87. Write the output:

```
double x[][] = {{5.3, 6.2, -1.1},
{2.9, 3.5, -6.7}};
System.out.print(x[0][1] * x[1][2] +
"\n" + "length of a row is: " + x[1].
length);
```

Ans : SQP 2010

```
-441.54
length of a row is: 3
```

88. Write the output of the below program segments:

(i)

```
static void main() {
   int x[][] = new int[6][50];
   int arr[][] = {{3, 9, 10, 20},
  {1,2, 3, 5}, {7, 9, 90, 100}};
  System.out.println(arr[1].
                            length);
  System.out.println(x[3].length);
  System.out.println(x[2][4length);
  }
```

(ii)

```
int y[] = new int[a];
System.out.println(y.length);
```

(iii)

```
int [][] x = new int[][];
int y[][] = new int[1][2];
System.out.println(y[1][1].length);
```

(iv)

```
int arr[] = new int[5];
System.out.println(arr[0]);
System.out.println(arr);
```

Ans : MAIN 2015

(i) 4
50
1

(ii) 97

(iii) 1

(iv) 0
0x6bcfe29

(Note: Junk characters depict the starting address of the array in memory.)

89. Write the statements to:

(i) Declare an array of strings with 3 rows and 5 columns.

(ii) Input the values of this array from the user in row-wise order.

(iii) Input values from the user in column-wise order.

Ans : COMP 2022

```
(i) String str[][] = new String[3]
   [5];
(ii) Scanner sc = new Scanner(System.
   in);
   System.out.println("Input
 values into the array in row-wise
   order:");
   for (int i = 0; i <= 2; i++) { //
Loop for Row
     for (int j = 0; j <= 4; j++) {
     // Loop for Column
         str[i][j] = sc.next();
    }
   }
(iii) Scanner sc = new Scanner(System.
   in);
   System.out.print("Input the
 values into the 2-dimension array");
 for (int i = 0; i <= 4; i++) { //
 Loop for Column
     for (int j = 0; j <= 2; j++) {
   // Loop for Row
         str[j][i] = sc.next();
    }
   }
```

90. String x[] = {"SAMSUNG", "NOKIA", "SONY", "MICROMAX", "BLACKBERRY"};
Give the output of the following statements:
(i) System.out.println(x[1]);
(ii) System.out.println(x[3].length());

Ans : SQP 2017

(i) NOKIA

(ii) 8

91. Consider the following String array and give the output

```
String arr[] = {"DELHI", "CHENNAI",
"MUMBAI", "LUCKNOW", "JAIPUR"};
System.out.println(arr[0].length() >
arr[3].length());
System.out.print(arr[4].
substring(0, 3));
```

Ans : COMP 2018

false
JAI

Download INDIA's Best Study app NODIA

Downloaded from www.pdf.tube

80. Write a Java statement to:
(i) Store the following names of countries and their capitals in a 2 × 2 matrix: INDIA, DELHI, NEPAL, KATHMANDU.
(ii) Store the following marks of two students in three subjects each in a double-dimension array:
(40, 80, 70) and (89, 79, 98)

Ans : SQP 2010

(i)
```
String c[][] = {{"INDIA", "DELHI"},
{"NEPAL", "KATHMANDU"}};
```
(ii)
```
int a[][] = {{40, 80, 70}, {89,
79, 98}};
```

81. How many locations are reserved for the following arrays?
(i) int x[ ] = new int[25];
(ii) float a[ ] = new float[42];

Ans : SQP 2023

(i) 25.
(ii) 42.

82. Write the output of the following segments of the program:

(i)
```
int arr[] = {2, 4, 6, 8, 10},{3,6
,9,12,15},{6,2,3,1,0};
for (int i = 0; i < 4; i++)
{
for (int j=0;j,<5;j++)
 System.out.print(arr[i][j]);
System.out.println();
}
```

(ii)
```
int arr2[][] = {{0, 1, 2}, {3, 4,
5}, {6, 7, 8}};
for (int i = 0; i < 3; i++) {
for (int j = 2; j >= 0; j--) {
System.out.print(arr2[i][j] + " ");
 }
 System.out.println();
 }
```

Ans : COMP 2013

(i) Output:
```
2 4 6 8 10
3 6 9 12 15
6 2 3 1 0
```
(ii) Output:
```
2 1 0
5 4 3
8 7 6
```

83. Show diagrammatically the address of each memory location of the array and the elements stored in them for the following arrays:
(i) int m[ ][ ] = {{1, 2, 3}, {9, 10, 11}, {15, 16, 17}};
(ii) String x[ ][ ] = {{"CALIFORNIA", "NEW YORK"}, {"WASHINGTON", "BOSTON"}};
(iii) char vow[ ][ ] = {{a, e, i, o, u}, {A, E, I, O, U}};

Ans : MAIN 2014

(i)

1	2	3
9	10	11
15	16	17

(ii)

CALIFORNIA	NEW YORK
WASHINGTON	BOSTON

(iii)

a	e	i	o	u
A	E	I	O	U

84. How many locations are reserved for the following arrays?
(i) int arr[ ][ ] = new int[5][2];
(ii) int r[ ][ ] = new int[3][3];
(iii) double x[ ][ ] = new double[3][6];

Ans : SQP 2011

(i) 10
(ii) 9
(iii) 18

85. What is the range of indices (subscripts) that the following arrays can have?
int ar[ ][ ] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};

Ans : COMP 2007

Row indices: 0 to 2
Column indices: 0 to 2

86. Write Java statements for the following:
Initialize -1 to an element in row 3 and column 5 of an array A[10][20].

Ans : MAIN 2008

```
A[2][4] = -1;
```

NODIA

TWO MARKS QUESTIONS

56. State the total size in bytes of the arrays a[4] of char data type and p[4] of float data type.

Ans : MAIN 2024

One char value takes 2 bytes, so a[4] takes 8 bytes. One float value takes 4 bytes, so p[4] takes 16 bytes.

57. Write one difference between Linear Search and Binary Search.

Ans : SQP 2019

(i) Binary search: Applied on sorted arrays and is faster for lengthy arrays.

(ii) Linear search: Works on unsorted arrays but is slower as it compares every element.

58. Write the difference between length and length() functions.

Ans : COMP 2012

length: Used to find the length of an array.
length(): A string function used to return the number of characters in a string.
Example:

```
int x[] = {10, -1, 3};
String y = "Absolutely";
int z = x.length;
int u = y.length();
Here, z = 3, u = 10.
```

59. Name the keyword that is used for allocating memory to an array.

Ans : MAIN 2016

new

60. If array[ ] = {1, 9, 8, 5, 2}:
(i) What is array.length?
(ii) What is array[2]?

Ans : SQP 2020

(i) 5

(ii) 8

61. Which element is num[9] of the array num?

Ans : COMP 2004

num[9] is the 10th element of the array num.

62. What is an array? Write a statement to declare an integer array of 10 elements.

Ans : MAIN 2006

An array is a finite set of adjacent memory locations that stores values of the same data type.

```
int x[] = new int[10];
```

63. What is the need for implementing the array concept in programming?

Ans : SQP 2010

A single variable can store only one value in its single memory location. To store multiple values, we need arrays. Arrays allow storing multiple elements under the same variable name with different subscripts, improving the quality of the program and reducing variables.

64. What is the difference between subscript and a subscripted variable?

Ans : COMP 2013

Subscripted variable: Provides access to multiple memory locations.
Subscript: Refers to the index number of an element in the array.
Example: In a[5], the subscript is 5.

65. What are the types of arrays?

Ans : MAIN 2014

Single-dimensional arrays, double-dimensional arrays, and multi-dimensional arrays.

66. What is the difference in the statements "Declaring an array" and "Accessing an array element"?

Ans : SQP 2011

Declaring an array: Specifies the data type and name of the array (e.g., int a[];).
Accessing an array element: Refers to a specific memory location (e.g., a[2] = 20;).

67. (i) What is the difference in the two Java statements given as:

```
int x[];
x = new int[10];
```

(ii) Write a single statement to perform the same job as above.

Ans : MAIN 2008

(i) The first statement declares the array, while the second allocates memory to the array.

(ii)

```
int x[] = new int[10];
```

68. Write a Java statement to store the first six multiples of 4 in a single subscripted variable var.

Click Here to Buy book on Amazon

92. String x[ ] = {"Artificial intelligence", "IOT", "Machine learning", "Big data"}
Give the output of the following statements:
(i) `System.out.println(x[3]);`
(ii) `System.out.println(x.length);`

Ans : MAIN 2019

(i) Big data
(ii) 4

FIFTEEN MARKS QUESTIONS

93. Write a program to input and store the weight of ten people. Sort and display them in descending order using the selection sort technique.

Ans : SQP 2024

```
import java.util.*;

public class Q5SortWeight {
     public static void main(String
args[]) {
         int weight[] = new int[10];
     Scanner buf = new Scanner(System.
in);

       System.out.println("Enter the
weight of 10 persons:");
       for (int i = 0; i <= 9; i++) {
          weight[i] = buf.nextInt();
        // Selection sorting
        int max = 0, index = 0;
       for (int i = 0; i <= 8; i++) {
            max = weight[i];
            for (int j = i + 1; j <=
9; j++) {
               if (weight[j] > max) {
                    max = weight[j];
                    index = j;
                }
            }
          weight[index] = weight[i];
            weight[i] = max;
        }

          System.out.println("SORTED
WEIGHTS=");
       for (int k = 0; k <= 9; k++) {
      System.out.println(weight[k]);
        }
    }
}
```

Variables used:
weight[ ]: Array to store weights,
i, j, k: Loop variables

94. Write a program to perform binary search on a list of integers given below, to search for an element input by the user. If it is found, display the element along with its position; otherwise, display the message "Search element not found".
5, 7, 9, 11, 15, 20, 30, 45, 89, 97

Ans : COMP 2003

```
import java.util.*;
class Binary {
    public static void main() {
         int x[] = {5, 7, 9, 11, 15,
20, 30, 45, 89, 97};
           System.out.println("Enter
element to search in the list");
     Scanner buf = new Scanner(System.
in);
        int elem = buf.nextInt();
       // Initialise lower bound and
upper bound
         int LB = 0, UB = 9; // 10
elements
        int mid = 0;
        boolean flag = false;
        do {
            mid = (LB + UB) / 2;
            if (elem == x[mid]) {
                          System.out.
println("Element is: " + elem);
           System.out.println("Found
at index: " + mid);
                          System.out.
println("Position in list: " + (mid
+ 1));
                flag = true;
                break;
          } else if (elem < x[mid]) {
                UB = mid - 1;
            } else {
                LB = mid + 1;
        } while (LB <= UB);
        if (flag == false) {
            System.out.print("Search
element not found");
        }
    }
```

Variables used :
x[] : Array of integers
LB : Lower Bound (index) of array from where to start the search

NODIA

Ans : MAIN 2017

```
import java.util.*;
class Merge {
    public static void main() {
        int P[] = new int[6];
        int Q[] = new int[4];
        int R[] = new int[10];
     Scanner buf = new Scanner(System.
in);
         System.out.println("Enter 6
elements of array P and 4 elements of
array Q:");
        for (int i = 0; i < 6; i++) {
            P[i] = buf.nextInt();
        for (int j = 0; j < 4; j++) {
            Q[j] = buf.nextInt();
         // Merge
         int k;
         for (k = 0; k < 6; k++) {
            R[k] = P[k];
         }
        for (int m = 0; m < 4; m++) {
            R[k] = Q[m];
            k = k + 1;
         }

         // Third array after merge
        for (int i = 0; i < 10; i++) {
           System.out.println(R[i]);
         }
    }
}
```

99. The annual examination results of 50 students in a class is tabulated as follows:

Roll No.	Subject A	Subject B	Subject C
...	...	...	...

Write a program to read the data, calculate, and display the following:

1. Average mark obtained by each student.
2. Print the roll number and average marks of the students whose average mark is above 80.
3. Print the roll number and average marks of the students whose average mark is below 40.

Ans : SQP 2021

```
import java.util.Scanner;

class Result {
    public static void main() {
     Scanner buf = new Scanner(System.
in);
        int roll[] = new int[50];
      double subA[] = new double[50];
      double subB[] = new double[50];
      double subC[] = new double[50];
       double avg[] = new double[50];
      System.out.println("Enter Roll
Numbers and marks for 3 subjects:");
       for (int i = 0; i < 50; i++) {
            System.out.println("Roll
Number:");
            roll[i] = buf.nextInt();

         System.out.println("Subject
A marks:");
          subA[i] = buf.nextDouble();

         System.out.println("Subject
B marks:");
          subB[i] = buf.nextDouble();

         System.out.println("Subject
C marks:");
          subC[i] = buf.nextDouble();

          avg[i] = (subA[i] + subB[i]
+ subC[i]) / 3;
        // Display results
            System.out.println("Roll
Number and Average Marks:");
       for (int i = 0; i < 50; i++) {
            System.out.println("Roll
No: " + roll[i] + ", Average: " +
avg[i]);
        }

      System.out.println("\nStudents
with Average Marks Above 80:");
       for (int i = 0; i < 50; i++) {
            if (avg[i] > 80) {
            System.out.println("Roll
No: " + roll[i] + ", Average: " +
avg[i]);
      System.out.println("\nStudents
with Average Marks Below 40:");
       for (int i = 0; i < 50; i++) {
            if (avg[i] < 40) {
            System.out.println("Roll
No: " + roll[i] + ", Average: " +
avg[i]);
            }
        }
    }
```

Downloaded from www.pdf.tube

```
a[i] <= 35000)
                d = 0.10 * a[i];
            else if (a[i] >= 35001 &&
a[i] <= 55000)
                d = 0.12 * a[i];
            else if (a[i] >= 55001 &&
a[i] <= 70000)
                d = 0.16 * a[i];
             else if (a[i] > 70000)
                d = 0.18 * a[i];
             net = a[i] - d;

              System.out.println((i +
1) + "\t" + n[i] + "\t" + a[i] + "\t"
+ d + "\t" + net);
        }
    }
}
```

Variables used:

n[ ]: Array to store names of customers
a[ ]: Array to store ticket amount for each customer
d[ ]: Array to store discount
net[ ]: Array to store net amount for each customer
j, i: Loop variables

97. Write a program to accept the names of 10 cities in a single dimension string array and their STD (Subscribers Trunk Dialing) codes in another single dimension integer array. Search for a name of a city input by the user in the list. If found, display "Search Successful" and print the name of the city along with its STD code, or else display the message "Search Unsuccessful, No such city in the list".

Ans : COMP 2012

```
import java.util.Scanner;
public class Ans9 {
    public static void main(String[]
args) {
      String city[] = new String[10];
        long std[] = new long [10];
     Scanner obj = new Scanner(System.
in);
system.out.println("Enter cities and
their corresponding std codes")
       // Input cities and their STD
codes
        for (int i = 0; i < 9; i++) {
           System.out.println("Enter
city name:");
            city[i] = obj.nextLine();
            System.out.println("Enter
STD code:");
             std[i] = obj.nextInt();
           sc.nextLine(); // consume
newline
        }

        // Search for a city
           System.out.println("Enter
city to search:");
       String search = sc.nextLine();
        boolean found = false;

       for (int j = 0; j < 10; j++) {
                        if (city[i].
equalsIgnoreCase(search)) {
                            System.out.
println("Search Successful");
                            System.out.
println("City: " + city[i] + ", STD
Code: " + std[j]);
                found = true;
                break;
            }
        }
        if (found==false) {
          System.out.println("Search
Unsuccessful, No such city in the
list.");
        }
    }
}
```

98. Write a program to store 6 elements in an array P, and 4 elements in an array Q and produce a third array R, containing all the elements of array P and Q. Display the resultant array.
Example:

INPUT		OUTPUT
P[ ]	Q[ ]	R[ ]
4	19	4
6	23	6
1	7	1
2	8	2
3		3
10		10
		19
		23
		7
		8

NODIA

```
        int a[] = {2, 5, 4, 1, 3};
         int min = a[0], max = a[0],
sum = 0;

        for (int i = 0; i < a.length;
i++) {
            if (min > a[i]) {
                min = a[i];
            }
            if (max < a[i]) {
                max = a[i];
            }
            sum += a[i];
        }

         System.out.println("Minimum
value: " + min);
         System.out.println("Maximum
value: " + max);
          System.out.println("Sum of
the elements: " + sum);
    }
}
```

103. Write a program to accept 15 integers from the keyboard, assuming that no integer entered is a zero. Perform selection sort on the integers and then print them in ascending order.

Ans : MAIN 2016

```
import java.util.*;
public class SelectionSorting {
     public static void main(String
args[]) {
        int arr[] = new int[15];
     Scanner buf = new Scanner(System.
in);

        System.out.println("Enter 15
nos.:");
       for (int i = 0; i < 15; i++) {
            arr[i] = buf.nextInt();
        }

        // Selection sort
       for (int i = 0; i < 14; i++) {
            int minIndex = i;
             for (int j = i + 1; j <
15; j++) {
                        if (arr[j] <
arr[minIndex]) {
                    minIndex = j;
                }
            }
             // Swap
            int temp = arr[minIndex];
             arr[minIndex] = arr[i];
             arr[i] = temp;
        }

          System.out.println("Sorted
numbers in ascending order:");
       for (int i = 0; i < 15; i++) {
         System.out.println(arr[i]);
        }
    }
}
```

104. Define a class to accept values into an array of double data type of size 20. Accept a double value from the user and search in the array using the linear search method. If the value is found, display the message "Found" with its position where it is present in the array. Otherwise, display the message "Not found".

Ans : SQP 2020

```
import java.util.Scanner;
class Main {
     public static void main(String
args[]) {
      Scanner sc = new Scanner(System.
in);
       double arr[] = new double[20];
        // Input Array
       for (int i = 0; i < 20; i++) {
           arr[i] = sc.nextDouble();
        }
        // Linear Search
        boolean flag = false;
         System.out.println("Enter a
number to search: ");
       double num = sc.nextDouble();
       for (int j = 0; j < 20; j++) {
            if (num == arr[j]) {
                flag = true;
           System.out.println("Found
at Index " + j);
            }
        }

        if (flag == false) {
             System.out.println("Not
found");
        }
    }
}
```

Click Here to Buy book on Amazon

```
}
```

Variables Used :
roll[ ]: Array of roll numbers of 50 students
subA[ ]: Array of marks obtained in subject A
subB[ ]: Array of marks obtained in subject B
subC[ ]: Array of marks obtained in subject C
avg[ ]: Array of the average marks obtained by 50 students
i, j, k: Loop variables used as subscripts for arrays

100. Write a program to input 10 integer elements in an array and sort them in descending order using the bubble sort technique.

Ans : SQP 2024

```
import java.util.Scanner;
public class BubbleSort {
    public static void main(String[]
args) {
        int a[] = new int[10];
    Scanner scan = new Scanner(System.
in);
        System.out.println("Enter 10
integers:");
       for (int i = 0; i < 10; i++) {
            a[i] = scan.nextInt();
        }

        // Bubble sort in descending
order
        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 9 -
i; j++) {
               if (a[j] < a[j + 1]) {
                    int temp = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = temp;
                }
            }
        }
          System.out.println("Sorted
Array in Descending Order:");
       for (int i = 0; i < 10; i++) {
           System.out.println(a[i]);
        }
    }
}
```

Variables Used:

a: Array of integer elements

k, i, j: Loop variables

t: Temporary variable used in the sorting process for swapping elements

101. Define a class and store the given city names in a single-dimensional array. Sort these names in alphabetical order using the Bubble Sort technique only.
INPUT Delhi, Bangalore, Agra, Mumbai, Kolkata
OUTPUT :Agra, Bangalore, Delhi, Kolkata, Mumbai

Ans : COMP 2023

```
class sortArray {
    public static void main() {
          String city[] = {"Delhi",
"Bangalore",    "Agra",    "Mumbai",
"Kolkata"};
        String temp;
        int i, j;
        int n = 5;

       for (i = 1; i <= n - 1; i++) {
            for (j = 0; j <= n - i -
1; j++) {
                        if ((city[j].
compareTo(city[j + 1])) > 0) {
                    temp = city[j];
                    city[j] = city[j
+ 1];
                city[j + 1] = temp;
               }
            }
        }

       System.out.println("Cities in
alphabetical order:");
        for (i = 0; i < n; i++) {
        System.out.println(city[i]);
        }
    }
}
```

Variables Used:
city[ ]: Stores list of cities
i, j, k: Loop variables
temp: Temporary string variable

102. Write a program to initialize the given data in an array and find the minimum and maximum values along with the sum of the given elements.

```
Numbers: 2, 5, 4, 1, 3
Output:  Minimum value: 1
             Maximum value: 5
             Sum of the elements: 15
```

Ans : SQP 2012

```
public class Aa {
    public static void main() {
```

Downloaded from www.pdf.tube

```
    }
}
```

Variables Used:
n[ ]: Array of 5 names
ph[ ]: Array of 5 corresponding phone numbers
f: Integer variable whose value changes if the condition is satisfied
m: Name whose phone number is to be searched
i: Loop variable

107. Define a class to declare an array to accept and store ten words. Display only those words which begin with the letter A or a and also end with the letter A or a.
EXAMPLE:
Input: Hari, Anita, Akash, Amrita, Alina, Devi, Rishab, John, Farha, AMITHA
Output:
Anita
Amrita
Alina
AMITHA

Ans : SQP 2022

```
import java.util.Scanner;

public class FilterWords {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
     String words[] = new String[10];
        System.out.println("Enter 10
words:");
       for (int i = 0; i < 10; i++) {
           words[i] = sc.nextLine();
        }
           System.out.println("Words
beginning and ending with A or a:");
       for (int i = 0; i < 10; i++) {
            String word = words[i];
           if ((word.startsWith("A")
|| word.startsWith("a")) &&
                   (word.endsWith("A")
|| word.endsWith("a"))) {
           System.out.println(word);
            }
       }
    }
```

108. Accept a list of 100 integers from the user and print their sum and average using a Java program.

Ans : COMP 2013

```
import java.util.*;
lass Q16 {
    public static void main(String[]
args) {
     Scanner OBJ = new Scanner(System.
in);
        double avg = 0.0;
        int sum = 0;
        int a[] = new int[100];
       for (int i = 0; i < 100; i++) {
           System.out.println("Enter
the value:");
            a[i] = OBJ.nextInt();
            sum = sum + a[i];
        }
        avg = sum / 100.0;
         System.out.println("Sum = "
+ sum);
         System.out.println("Average
= " + avg);
    }
}
```

109. WAP in Java that accepts 20 numbers from the user and prints only the even numbers together.

Ans : MAIN 2014

```
import java.util.*;
class Q17 {
    public static void main(String[]
args) {
     Scanner OBJ = new Scanner(System.
in);
        int m = 0;
        int b[] = new int[20];
        int a[] = new int[20];
       for (int i = 0; i < 20; i++) {
           System.out.println("Enter
the value:");
            a[i] = OBJ.nextInt();
            if (a[i] % 2 == 0) {
                b[m] = a[i];
                m++;
            }
        }

       for (int k = 0; k < m; k++) {
            System.out.print(b[k] +
" ");
        }
    }
}
```

Download INDIA's Best Study app NODIA

105. Write a program to accept the year of graduation from school as an integer value from the user. Using the Binary Search technique on the sorted array of integers given below, output the message "Record exists" if the value input is located in the array. If not, output the message "Record does not exist".
Array:
{1982, 1987, 1993, 1996, 1999, 2003, 2006, 2007, 2009, 2010}

Ans : COMP 2004

```
import java.util.Scanner;
public class GraduationSearch {
    public static void main(String[]
args) {
          int years[] = {1982, 1987,
1993, 1996, 1999, 2003, 2006, 2007,
2009, 2010};
     Scanner sc = new Scanner(System.
in);

       System.out.println("Enter the
year of graduation to search:");
      int searchYear = sc.nextInt();

        // Binary Search
       int lb = 0, ub = years.length
- 1;
        boolean found = false;

        while (lb <= ub) {
            int mid = (lb + ub) / 2;

                  if (years[mid] ==
searchYear) {
                found = true;
                break;
             } else if (years[mid] <
searchYear) {
                lb = mid + 1;
            } else {
                ub = mid - 1;
            }
        }

        if (found) {
          System.out.println("Record
exists");
        } else {
          System.out.println("Record
does not exist");
        }
    }
}
```

Variables Used:
lb, ub: Lower bound and upper bound of the array/list
mid: Middle index
flag: To identify if the record exists/does not exist in the list

106. Write a program to initialize an array of 5 names and initialize another array with their respective telephone numbers. Search for a name input by the user, in the list. If found, display "Search Successful" and print the name along with the telephone number, otherwise display "Search unsuccessful. Name not enlisted".

Ans : MAIN 2006

```
import java.util.Scanner;
public class NameSearch {
    public static void main(String[]
args) {
          String names[] = {"Alice",
"Bob", "Charlie", "Diana", "Edward"};
            String phoneNumbers[] =
{"1234567890",          "9876543210",
"1122334455",          "9988776655",
"7766554433"};

     Scanner sc = new Scanner(System.
in);
       System.out.println("Enter the
name to search:");
    String searchName = sc.nextLine();

        boolean found = false;

         for (int i = 0; i < names.
length; i++) {
                    if (names[i].
equalsIgnoreCase(searchName)) {
                found = true;
                        System.out.
println("Search Successful");
           System.out.println("Name:
" + names[i] + ", Phone Number: " +
phoneNumbers[i]);
                break;
            }
        }
        if (!found) {
          System.out.println("Search
unsuccessful. Name not enlisted.");
        }
```

Click Here to Buy book on Amazon

```
                leapYearCount++;
            }
        }
          System.out.println("Number of leap years inputted: " + leapYearCount);
    }
}
```

114. Write a program to accept a list of 50 integers from the user and another value to search in the list. Your program should print the position of the value in the list.

Ans : COMP 2022

```
import java.util.*;
class Q22 {
    public static void main(String[] args) {
     Scanner obj = new Scanner(System.in);
        int pos = -1;
           System.out.println("Enter the value whose position is to be searched:");
        int v = obj.nextInt();
        int a[] = new int[50];
       for (int i = 0; i < 50; i++) {
           System.out.println("Enter the number:");
            a[i] = obj.nextInt();
        }
       for (int i = 0; i < 50; i++) {
            if (v == a[i]) {
                pos = i + 1;
                break;
            }
        }
        if (pos == -1) {
           System.out.println("Value not found in the list.");
        } else {
        System.out.println("Position = " + pos);
        }
    }
}
```

115. WAP in Java to enter a year and print the total number of days in that year, taking the case of the leap year which has 29 days in February. Your program should make use of an array to store the days of each month.

Ans : MAIN 2015

```
import java.util.*;
class Q23 {
    public static void main(String[] args) {
     Scanner obj = new Scanner(System.in);
        int sum = 0;
         int a[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; // Non-leap year
         int b[] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; // Leap year
         System.out.println("Enter a year:");
        int n = obj.nextInt();
       if (n % 4 == 0 && (n % 100 != 0 || n % 400 == 0)) {
            for (int i = 0; i < 12; i++) {
                sum = sum + b[i];
        } else {
            for (int i = 0; i < 12; i++) {
                sum = sum + a[i];
            }
       System.out.println("Total no. of days = " + sum);
    }
}
```

116. WAP in Java to accept two arrays of 10 integers and 20 integers respectively and finally store the sum of corresponding elements in a third array. Display the resultant array.

Ans : SQP 2018

```
import java.util.*;
class Q24 {
    public static void main(String[] args) {
     Scanner obj = new Scanner(System.in);
        int c[] = new int[20];
        int a[] = new int[10];
        int b[] = new int[20];
        // Input elements of array a
       for (int i = 0; i < 10; i++) {
           System.out.println("Enter the elements of array a:");
            a[i] = obj.nextInt();
        }
        // Input elements of array b
       for (int i = 0; i < 20; i++) {
```

NODIA

110. WAP in Java that stores the first 10 natural numbers in a single subscripted variable and then prints them in reverse order.

Ans : SQP 2011

```
class Q18 {
    public static void main(String[]
args) {
        int a[] = new int[10];

       for (int i = 0; i < 10; i++) {
            a[i] = i + 1;
        }
       for (int j = 9; j >= 0; j--) {
           System.out.println(a[j]);
        }
    }
```

111. Initialize an array num[] by the following values:
3.52, 5.96, 2.9, 1.06, 4.28, 3.13
WAP in Java that rounds off each number to the nearest integer.

Ans : COMP 2007

```
class Q19 {
public   static   void   main(String
args[]) {
int m,y;
double n;
double        num[]        =        {
3.52,5.96,2.9,1.M,4.28,3.13}, ;
for(int i=0;i<6;i+ +) {
6=(int) num[i];
n=num[i]-m;
i(n>=0.s) {
y=m+ 1;
System.out.println(y);
)
else System.out.println(m);
   }
  }
}
```

112. Write a program to input twenty names in an array. Arrange these names in descending order of alphabets, using the bubble sort technique.

Ans : MAIN 2015

```
import java.util.Scanner;
public class ArrangeNames {
    public static void main(String[]
args) {
        String a[] = new String[20];
    Scanner scan = new Scanner(System.
in);
          System.out.println("Please
enter the twenty names: ");
       for (int i = 0; i < 20; i++) {
            a[i] = scan.next();
        }
       // Bubble Sort for Descending
Order
       for (int i = 0; i < 19; i++) {
            for (int j = 0; j < 19 -
i; j++) {
                            if  (a[j].
compareToIgnoreCase(a[j + 1]) < 0) {
                  String temp = a[j];
                     a[j] = a[j + 1];
                     a[j + 1] = temp;
                }
            }
        }
        System.out.println("Names in
descending order:");
       for (int i = 0; i < 20; i++) {
           System.out.println(a[i]);
        }
    }
}
```

Variables Used:
i, j, k: Loop variables
a: List of 20 names
t: A string

113. Initialize an array year[] with the values inputted from the keyboard. WAP that prints the number of leap years inputted.

Ans : SQP 2010

```
import java.util.Scanner;
public class LeapYearCount {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
        int year[] = new int[10]; //
Array for storing years
        int leapYearCount = 0;
        System.out.println("Enter 10
years:");
       for (int i = 0; i < 10; i++) {
            year[i] = sc.nextInt();
             // Check if the year is
a leap year
            if ((year[i] % 4 == 0 &&
year[i] % 100 != 0) || (year[i] % 400
== 0)) {
```

Downloaded from www.pdf.tube

UB : Upper Bound (index) for array, till where to search
flag : To test whether the element is found or not
mid : The middle index, between LB and UB. The element in the array at the middle index is compared with the element to be searched

95. Define a class to perform binary search on a list of integers given below, to search for an element input by the user. If it is found, display the element along with its position; otherwise, display the message "Search element not found".
2, 5, 7, 10, 15, 20, 29, 30, 46, 50

Ans : MAIN 2002

```
import java.util.*;
class Main {

     public static void main(String
args[]) {

      Scanner sc = new Scanner(System.
in);
         int a[] = {2, 5, 7, 10, 15,
20, 29, 30, 46, 50};

          System.out.println("Enter a
number to search in the list: ");
        int num = sc.nextInt();
        int lb = 0;
        int ub = 9;
        int mid = (lb + ub) / 2;

        boolean flag = false;
        while (lb <= ub) {
            if (num == a[mid]) {

              System.out.println(num
+ " is found at index " + mid);
                flag = true;
                break;
           } else if (num < a[mid]) {
                ub = mid - 1;
            } else {
                lb = mid + 1;
            }
            mid = (lb + ub) / 2;
        }

        if (flag == false)
          System.out.println("Search
element not found");
        }
    }
```

Download INDIA's Best Study app NODIA

NODIA

96. Shasha Travels Pvt. Ltd. gives the following discount to its customers:

Ticket Amount	Discount
Above 70000	18%
55001 to 70000	16%
35001 to 55000	12%
25001 to 35000	10%
Less than 25001	2%

Write a program to input the name and ticket amount for the customer and calculate the discount amount and net amount to be paid. Display the output in the following format for each customer:

S. No	Name	Ticket Charges	Discount	Net Amount
1	...	...	...	...

Assume that there are 15 customers, first customer is given the serial no (Sl. No.) 1, next customer 2 ... and so on.

Ans : SQP 2001

```
import java.util.*;
class Shasha {
    public static void main() {
     Scanner buf = new Scanner(System.
in);
        String n[] = new String[15];
        Long a[] = new Long[15];
        double net, d;
        d = 0.0;
         for (int i = 0; i < 2; i++)
{ // loop for input
            System.out.println("Enter
name ");
             n[i] = buf.nextLine();
            System.out.println("Enter
amount ");
          a[i] = Long.parseLong(buf.
nextLine());
        }

        System.out.println("Sl. No.\
tName\tTicket charges\tDiscount\tNet
amount");
         for (int i = 0; i < 2; i++)
{ // loop for calculation
            if (a[i] < 25001)
                d = 0.02 * a[i];
           else if (a[i] >= 25001 &&
```

```
            System.out.println("Enter
the elements of array b:");
                b[i] = obj.nextInt();
            }
            // Combine both arrays
           for (int i = 0; i < 10; i++) {
                c[i] = a[i] + b[i];
            }
           for (int i = 10; i < 20; i++) {
                c[i] = b[i];
            }
           // Display the resultant array
           for (int i = 0; i < 20; i++) {
               System.out.println(c[i]);
            }
    }
```

117. Two single-dimension arrays contain the elements as follows:
x[10] = {5, -3, -2, 1, 0, 12, 14, 16, 25, 13}
y[8] = {6, 5, 10, 15, 18, 20, 22, 30}
Write a program that results in a third array z as follows:
z[18] = {5, 6, -3, 5, -2, 10, 1, 15, 0, 18, 12, 20, 14, 22, 16, 30, 25, 13}

Ans : COMP 2019

```
class Q25 {
    public static void main(String[]
args) {
         int x[] = {5, -3, -2, 1, 0,
12, 14, 16, 25, 13};
        int y[] = {6, 5, 10, 15, 18,
20, 22, 30};
        int z[] = new int[18];
        int j = 0;
        for (int i = 0; i < 16; i +=
2) {
            z[i] = x[j];
            j++;
        }
int k=0;
for(int i= 1;i< 16;i+ =2) {
z[i]=y[k];
k+ +;
        z[16] = x[8];
        z[17] = x[9];
       for (int i = 0; i < 18; i++) {
            System.out.print(z[i] +
" ");
        }
    }
}
```

118. Initialize a one-dimensional array with the first ten natural numbers and then display them in the following format:
Format:
1 2 3
4 5 6
7 8 9
10

Ans : MAIN 2009

```
class Q26 {
    public static void main(String[]
args) {
        int a[] = {1, 2, 3, 4, 5, 6,
7, 8, 9, 10};
        int k = 0;
       for (int i = 1; i <= 3; i++) {
            for (int j = k; j <= k +
2; j++) {
                System.out.print(a[j]
+ " ");
            }
            System.out.println();
            k += 3;
        }
        System.out.println(a[9]);
    }
}
```

119. WAP to print the product of all odd subscripted values of an array arr[25].

Ans : SQP 2003

```
import java.util.*;
class {
    public static void main(String[]
args) {
    Scanner obj = new Scanner(System.
in);
        int p = 1;
        int arr[] = new int[25];
       for (int i = 0; i < 25; i++) {
           System.out.println("Enter
value for position " + i + ":");
            arr[i] = obj.nextInt();
        }
        for (int i = 1; i < 25; i +=
2) { // Odd subscripted indices
            p *= arr[i];
        }
         System.out.println("Product
of all odd subscripted values: " +
p);
    }
}
```

Click Here to Buy book on Amazon

128. WAP to process the exam scores for a group of students in a course. Assume that students have to appear in the same number of tests. Enter each students name and the exam scores, calculate the average score for each student and an overall class average.

Ans : SQP 2016

```
import java.util.*;
public class {
     public static void main(String
args[]) {
        int arr[] = new int[10];
     Scanner buf = new Scanner(System.
in);

       System.out.println("Enter the
number of students: ");
        int num = buf.nextInt();

       System.out.println("Enter the
number of tests: ");
        int t = buf.nextInt();

        float s = 0;

          for (int i = 1; i <= num;
i++) {
            float tot = 0;
          System.out.println("Please
enter the students name:");
            String nam = buf.next();

              for (int j = 0; j < t;
j++) {
          System.out.println("Please
enter the test marks:");
             int m1 = buf.nextInt();
               tot = tot + m1;
            }

             System.out.println("The
average score of " + nam + " is " +
tot / t);
            s = s + tot;
        }

       System.out.println("The class
average is: " + s / num);
    }
}
```

Download INDIA's Best Study app NODIA

NODIA

129. Use arrays in Java and WAP to store roll numbers, names, and marks of six subjects for 10 students. Calculate the percentage marks obtained by each. Assume the maximum marks in each subject as 100. Calculate the grade as per the given criteria:

Percent	Grade
80 to 100	A
60 to 79	B
40 to 59	C
Less than 40	D

Ans : SQP 2020

```
import java.util.*;
public class {
     public static void main(String
args[]) {
    Scanner buf = new Scanner(System.
in);

       String nam[] = new String[20];
         int r[] = new int[20];
         int mk[] = new int[6];
char grade= ;
         int t;

        for (int i = 0; i < 10; i++) {
            System.out.println("Enter
your name and roll number:");
             nam[i] = buf.nextLine();
             r[i] = buf.nextInt();
             t = 0;

            System.out.println("Enter
the marks in 6 subjects:");
              for (int j = 0; j < 6;
j++) {
               mk[j] = buf.nextInt();
                 t += mk[j];
             }

             double p = t / 6.0;
System.out.println("The   per   cent
marks obtained by "+nam[i]+ " is
"+p+"%");
             char grade;

           if (p >= 80 && p <= 100) {
                grade = A;
            } else if (p >= 60 && p
<= 79) {
                grade = B;
```

```
corresponding roll number
        for (int i = 0; i < 35; i++) {
             if (T[i] > max) {
                 max = T[i];
                 roll = R[i];
             }
         }
         // Print the result
          System.out.println("Maximum
total marks = " + max);
         System.out.println("Roll no.
of the student who scored it = " +
roll);
    }
}
```

123. The annual exam result of 50 students of a class X-D of St. Xaviers is tabulated as follows:
Roll No. Subject A Subject B Subject C
WAP in Java to read the data and determine:
(i) Total marks obtained by each student.
(ii) Highest marks in each subject and the roll number of the student who obtained it.
(iii) Highest total marks and the roll number of the student who got it.

Ans : COMP 2017

```
import java.util.*;
class {
    public static void main(String[]
args) {
          int R[] = new int[50];  //
Array for roll numbers
          int s1[] = new int[50]; //
Marks in Subject A
          int s2[] = new int[50]; //
Marks in Subject B
          int s3[] = new int[50]; //
Marks in Subject C
         int tot[] = new int[50]; //
Total marks for each student
     Scanner obj = new Scanner(System.
in);
          // Input roll numbers and
marks
        for (int i = 0; i < 50; i++) {
            System.out.println("Enter
the roll number of the student:");
            R[i] = obj.nextInt();
           System.out.println("Enter
the marks of three subjects:");
            s1[i] = obj.nextInt();
            s2[i] = obj.nextInt();
            s3[i] = obj.nextInt();
            tot[i] = s1[i] + s2[i] +
s3[i];
             System.out.println("The
total marks of roll no. " + R[i] + "
is " + tot[i]);
         }
         // Initialize variables for
highest marks
         int Max = tot[0];
        int Hs1 = s1[0], Hs2 = s2[0],
Hs3 = s3[0];
         int rt = 0, rs1 = 0, rs2 =
0, rs3 = 0;
          // Determine highest marks
and roll numbers
        for (int i = 0; i < 50; i++) {
            if (tot[i] > Max) {
                Max = tot[i];
                rt = R[i];
            }
            if (s1[i] > Hs1) {
                Hs1 = s1[i];
                rs1 = R[i];
            }
            if (s2[i] > Hs2) {
                Hs2 = s2[i];
                rs2 = R[i];
            }
            if (s3[i] > Hs3) {
                Hs3 = s3[i];
                rs3 = R[i];
            }
         }
         // Print the results
        System.out.println("The total
highest is = " + Max + " obtained by
roll number " + rt);
             System.out.println("The
highest in subject 1 = " + Hs1 + "
obtained by roll number " + rs1);
             System.out.println("The
highest in subject 2 = " + Hs2 + "
obtained by roll number " + rs2);
             System.out.println("The
highest in subject 3 = " + Hs3 + "
obtained by roll number " + rs3);
    }
```

124. Using the technique of Binary Searching, print the position of a value in an arranged list of 5 integers. Assume that the entered list is in ascending order.

Ans : MAIN 2021

NODIA

120. WAP to count the number of prime numbers stored in an integer array of size 25.

Ans : COMP 2002

```
import java.util.*;
class {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int count = 0;
        int a[] = new int[25];
        // Input elements
       for (int i = 0; i < 25; i++) {
           System.out.println("Enter
the number:");
            a[i] = obj.nextInt();
        }
        // Check for prime numbers
       for (int i = 0; i < 25; i++) {
            int c = 0;
           for (int j = 1; j <= a[i];
j++) {
                if (a[i] % j == 0) {
                    c++;
                }
            }
            if (c == 2) {
                count++;
            }
        }
       System.out.println("Number of
prime numbers: " + count);
    }
}
```

121. WAP to generate and store the 30 terms of Fibonacci list of the following numbers in an array fibArr[30]:
5, 8, 13, 21, 34, ...
Finally, print the sum of all elements contained in fibArr[30].

Ans : MAIN 2001

```
class {
    public static void main(String[]
args) {
        int fibArr[] = new int[30];
        int a = 5, b = 8;
        int sum = 0;

         // Initialize the first two
terms
        fibArr[0] = a;
        fibArr[1] = b;

          // Generate the Fibonacci
series
       for (int i = 2; i < 30; i++) {
            fibArr[i] = fibArr[i - 1]
+ fibArr[i - 2];
        }

         // Calculate the sum of all
terms
       for (int i = 0; i < 30; i++) {
            sum += fibArr[i];
        }

        // Output the sum
          System.out.println("Sum of
Fibonacci terms: " + sum);
    }
}
```

122. WAP to store lists of roll numbers and total marks of 35 students of a class. Print the maximum total marks and the roll number of the student who got it.

Ans : SQP 2005

```
import java.util.*;
class {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);

          int T[] = new int[35]; //
Array to store total marks
          int R[] = new int[35]; //
Array to store roll numbers
         int max = 0;             //
Variable to store maximum marks
         int roll = 0;            //
Variable to store roll number of the
student with maximum marks
          // Input roll numbers and
marks
       for (int i = 0; i < 35; i++) {
           System.out.println("Enter
the roll number of the student:");
            R[i] = obj.nextInt();
           System.out.println("Enter
the total marks of the student:");
            T[i] = obj.nextInt();
        }
          // Find maximum marks and
```

Downloaded from www.pdf.tube

```
public class SelectionSorting {
     public static void main(String
args[]) {
        int arr[] = new int[20];
     Scanner buf = new Scanner(System.
in);
         System.out.println("Enter 5
numbers: ");
        for (int i = 0; i < 5; i++) {
            arr[i] = buf.nextInt();
        }
        int n = 5;
        int min, minPosition = 0;
        // Selection Sort
         for (int i = 0; i < n - 1;
i++) {
            min = arr[i];
             for (int j = i + 1; j <
n; j++) {
                if (arr[j] < min) {
                    min = arr[j];
                    minPosition = j;
                }
            }

             // Swap minimum element
with the current element
            int temp = arr[i];
          arr[i] = arr[minPosition];
            arr[minPosition] = temp;
        }

        // Display the sorted array
          System.out.println("Sorted
array in ascending order:");
        for (int i = 0; i < n; i++) {
             System.out.print(arr[i]
+ " ");
        }
    }
}
```

(ii)

```
import java.util.*;
public class BubbleSorting {
     public static void main(String
args[]) {
        int arr[] = new int[20];
     Scanner buf = new Scanner(System.
in);
        System.out.println("Enter 15
numbers: ");
       for (int i = 0; i < 15; i++) {
            arr[i] = buf.nextInt();
        }
        int n = 15;
        // Bubble Sort
         for (int i = 0; i < n - 1;
i++) {
             for (int j = 0; j < n -
1 - i; j++) {
                 if (arr[j] > arr[j +
1]) {
                     // Swap adjacent
elements
                     int t = arr[j];
                       arr[j] = arr[j
+ 1];
                     arr[j + 1] = t;
                 }
             }
        }
        // Display the sorted array
             System.out.println("The
numbers in ascending order are: ");
       for (int k = 0; k < n; k++) {
         System.out.println(arr[k]);
        }
    }
}
```

127. WAP to store 10 numbers in an array. Create another array that contains the squared values of all elements of the first array. Display the resulting array.

Ans : MAIN 2012

```
import java.util.*;
public class {
     public static void main(String
args[]) {
    Scanner inp = new Scanner(System.
in);
        int arr[] = new int[10];

          System.out.println("Please
enter 10 numbers:");
       for (int i = 0; i < 10; i++) {
            arr[i] = inp.nextInt();
        }

        int b[] = new int[10];
       for (int i = 0; i < 10; i++) {
            b[i] = arr[i] * arr[i];
          System.out.println("Square
of " + arr[i] + " is " + b[i]);
        }
    }
}
```

NODIA

```
import java.util.*;
class BinarySearch {
     public static void main(String
args[]) {
        int lb, ub, mid;
        int arr[] = new int[10];
     Scanner buf = new Scanner(System.
in);
         System.out.println("Enter 5
numbers in ascending order:");
       for (int i = 0; i < 5; i++) {
           arr[i] = buf.nextInt();
        }
        ub = 4;
        lb = 0;
         System.out.println("Enter a
number to search in the list:");
        int value = buf.nextInt();
        mid = (lb + ub) / 2;
         while (value != arr[mid]) {
// To find the middle element
            if (value < arr[mid]) {
                ub = mid;
                 } else if (value >
arr[mid]) {
                lb = mid;
            }
            mid = (lb + ub) / 2;
        }
        System.out.println("Position
is: " + (mid + 1));
    }
}
```

125. Write a program to initialize the Seven Wonders of the World along with their locations in two different arrays. Search for a name of the country input by the user. If found, display the name of the country along with its Wonder; otherwise, display "Sorry Not Found!".

Seven wonders: CHICHEN ITZA, CHRIST THE REDEEMER, TAJMAHAL, GREAT WALL OF CHINA, MACHU PICCHU, PETRA, COLOSSEUM

Locations : MEXICO, BRAZIL, INDIA, CHINA, PERU, JORDAN, ITALY

Example:

Country Name: INDIA

Output: INDIA - TAJMAHAL

Country Name: USA

Output: Sorry Not Found!

Ans : SQP 2016

```
import java.util.*;
public class  {
     public static void main(String
args[]) {
     String won[] = {"CHITCHEN ITZA",
"CHRIST  THE  REDEEMER",  "TAJMAHAL",
"GREAT WALL OF CHINA", "MACHU PICCHU",
"PETRA", "COLOSSEUM"};
          String loc[] = {"MEXICO",
"BRAZIL", "INDIA", "CHINA", "PERU",
"JORDAN", "ITALY"};

        System.out.println("The list
of country name is:");
          for (int i = 0; i < loc.
length; i++) {
         System.out.println(loc[i]);
        }

     Scanner sc = new Scanner(System.
in);
           System.out.println("Enter
country name:");
        String country = sc.next();
        int f = 0;
         String country1 = country.
toUpperCase();

          for (int i = 0; i < loc.
length; i++) {
      if (country1.compareTo(loc[i])
== 0) {
           System.out.println(loc[i]
+ " - " + won[i]);
                f = 1;
            }
        }

        if (f == 0) {
           System.out.println("Sorry
not found!");
        }
    }
}
```

126. WAP to accept 5 integers from the console. Sort the list in ascending order using:

(i) Selection Sort Method

(ii) Bubble Sorting Method

Ans : COMP 2023

(i)

```
import java.util.*;
```

Click Here to Buy book on Amazon

```
            } else if (p >= 40 && p
<= 59) {
                grade = C;
            } else {
                grade = D;
            }

             System.out.println("The
grade is " + grade);
        }
    }
}
```

130. WAP to print the second largest number from a list of 35 numbers and print the number as well as its position where it occurs.

Ans : COMP 2004

```
import java.util.*;
public class {
     public static void main(String
args[]) {
        int arr[] = new int[40];
        int a[] = new int[40];
     Scanner buf = new Scanner(System.
in);

        System.out.println("Enter 35
numbers:");
       for (int i = 0; i < 35; i++) {
            arr[i] = buf.nextInt();
            a[i] = arr[i];
        }

        int n = 35;
         for (int i = 0; i < n - 1;
i++) {
             for (int j = 0; j < n -
i - 1; j++) {
                 if (arr[j] > arr[j +
1]) {
                     int t = arr[j];
                       arr[j] = arr[j
+ 1];
                     arr[j + 1] = t;
                 }
             }
        }

      System.out.println("The second
largest number is " + arr[33]);
        for (int k = 0; k < n; k++) {
            if (arr[33] == a[k]) {
                    System.out.
println("Position where second
largest number exists in original
list is: " + k);
            }
        }
    }
```

131. WAP in Java that reads the following list of countries and their respective cities into arrays. The program should accept the name of a country in the list as input and print the corresponding city name as output. The program should give an error message when a city is asked or a country whose name is not in the list.

Ans : MAIN 2006

```
import java.util.*;

class {
    public static void main(String[]
args) {
         String count[] = {"India",
"China",     "Japan",     "Russia",
"Srilanka"};
          String city[] = {"Delhi",
"Beijing",    "Tokyo",    "Moscow",
"Colombo"};
        boolean v = false;

          System.out.println("Enter
the country whose city you want to
know:");
     Scanner obj = new Scanner(System.
in);
        String x = obj.nextLine();

        for (int i = 0; i < 5; i++) {
            if (x.compareTo(count[i])
== 0) {
                v = true;
             System.out.println("City
is " + city[i]);
            }
        }

        if (v == false) {
         System.out.println("Country
not found");
        }
    }
}
```

NODIA

Download INDIA's Best Study app NODIA

```
        int x[][] = {{6, 9, 0, 8},
{3, 2, 1, 5}, {2, 9, 0, 1}, {4, 1,
9, 6}};
       System.out.println("1. Sum of
main diagonal elements");
       System.out.println("2. Sum of
right diagonal elements");
      char c = buf.next().charAt(0);
       switch (c) {
           case 1:
               int s = 0;
               for (int i = 0; i <=
3; i++) {
                   s = s + x[i][i];
               }
             System.out.println(s);
               break;
           case 2:
               int r = 0;
               int j = 3;
               for (int i = 0; i <=
3; i++) {
                   r = r + x[i][j];
                   j--;
               }
             System.out.println(r);
         break;
default:System.out.prinfln(\ilrong
choice entered");
       }
   }
}
```

137. Write a program in Java to compute the
(i) product of all elements below the main diagonal,
(ii) sum of all elements above the main diagonal of a matrix of the order 4×4.

Ans : MAIN 2008

```
import java.util.*;
class {
    public static void main(String[]
args) {
     Scanner Obj = new Scanner(System.
in);
        int x[][] = new int[4][4];

       for (int i = 0; i < 4; i++) {
          System.out.println("Enter
the elements of " + i + " row");
            for (int j = 0; j < 4;
j++) {
                    x[i][j] = Obj.
nextInt();
            }
        }

         System.out.println("Enter 1
to find the product and enter 2 to find
the sum");
        int c = Obj.nextInt();
          int p = 1; // For product
below the diagonal
        int s = 0; // For sum above
the diagonal

        if (c == 1) {
             for (int i = 1; i < 4;
i++) {
                 for (int j = 0; j <
i; j++) {
                     p *= x[i][j];
                 }
             }
             System.out.println("The
product of elements below the main
diagonal is: " + p);
        } else if (c == 2) {
             for (int i = 0; i < 4;
i++) {
                 for (int j = i + 1;
j < 4; j++) {
                     s += x[i][j];
                 }
             }
          System.out.println("The sum
of elements above the main diagonal
is: " + s);
        } else {
         System.out.println("Invalid
choice");
        }
    }
}
```

138. If two matrices A and B are given as follows:

$$A = \begin{bmatrix} 2 & 1 \\ 5 & 6 \end{bmatrix}, B = \begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}$$

WAP to compute the sum of the two matrices and display the resulting matrix.

Ans : SQP 2010

```
import java.util.*;
class {
    public static void main(String[]
args) {
```

Downloaded from www.pdf.tube

134. WAP to perform the following tasks on a double subscripted variable A[x][y]:
(i) Compute the sum of all elements of the array.
(ii) Print all odd numbers of the array.

Ans : MAIN 2014

```
import java.util.*;
public class {
     public static void main(String
args[]) {
        int x = 2, y = 3;
        int a[][] = new int[x][y];
      Scanner b = new Scanner(System.
in);
          System.out.println("Please
enter the numbers: ");
        for(int i = 0; i < x; i++) {
              for(int j = 0; j < y;
j++) {
              a[i][j] = b.nextInt();
            }
        }

        int s = 0;
        for(int i = 0; i < x; i++) {
              for(int j = 0; j < y;
j++) {
                s += a[i][j];
              if(a[i][j] % 2 != 0) {
                          System.out.
println("Odd number in the list is:
" + a[i][j]);
                }
            }
        }
          System.out.println("Sum of
all elements is: " + s);
    }
}
```

135. Accept an array of 3 rows and 4 columns from the console.
(i) Print the sum of elements of each row of the array.
(ii) Print the sum of elements of each column of this array.

Ans : SQP 2011

```
import java.util.*;
public class {
     public static void main(String
args[]) {
        int a[][] = new int[3][4];
      Scanner b = new Scanner(System.
in);

        // Input the elements of the
array
        for(int i = 0; i < 3; i++) {
              for(int j = 0; j < 4;
j++) {
               a[i][j] = b.nextInt();
            }
        }

        // Compute and print the sum
of elements of each row
        for(int i = 0; i < 3; i++) {
            int sr = 0;
              for(int j = 0; j < 4;
j++) {
                sr += a[i][j];
            }
             System.out.println("Sum
of elements of row " + (i + 1) + "
is: " + sr);
        }

System.out.println();

for (int k = 0; k < 4; k++) {
    int sc = 0;
    for (int j = 0; j < 3; j++) {
        sc += a[j][k];
    }
     System.out.println("The sum of
column number " + k + " is " + sc);
}
```

136. Consider the following matrix:

6	9	0	8
3	2	0	1
4	1	9	6

Write a program to perform the following tasks on the given matrix depending on the choice entered by the user:
(i) Sum of main diagonal elements.
(ii) Sum of right diagonal elements.

Ans : COMP 2007

```
import java.util.*;
class {
    public static void main(String[]
args) {
     Scanner buf = new Scanner(System.
in);
```

NODIA

132. A dealer sells (1) Badminton rackets (2) Shuttlecocks (3) Nets in boxes containing 10 each. WAP to create a bill using the information given below:
Input: Date of purchase, name of buyer, price of each item, quantity of each item.

Output	Billing Details				
	S. no.	Item	Price	Quantity	Amount

Add 8% sales tax and print the net amount to be paid.

Ans : SQP 2010

```
import java.util.*;
class  {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
       String item[] = new String[3];
      double price[] = new double[3];
        String date;
        int qty[] = new int[3];
       double amt[] = new double[3];

           System.out.println("Enter
date of purchase");
        date = obj.nextLine();

           System.out.println("Enter
your name");
        String nam = obj.nextLine();

        double s = 0, sum = 0;
for(int i=0; i<3; i++) {
      System.out.println("Enter  the
article");
    item[i] = obj.nextLine();
       System.out.println("Price  of
item");
    price[i] = obj.nextDouble();
     System.out.println("Quantity of
item");
    qty[i] = obj.nextInt();
}
amt[i] = qty[i] * price[i];
s = s + amt[i];
}
sum = s * (8.0 / 100) + s;
System.out.println("BILLING
DETAILS");
System.out.println("S.No. Item Price
Quantity Amount");
for(int k = 0; k <= 2; k++) {
    int p = k + 1;
      System.out.println(p  +  "  "  +
item[k]  +  "  "  +  price[k]  +  "  "  +
qty[k] + " " + amt[k]);
}
System.out.println("Net amount after
adding 8% sales tax = " + sum);
          }
     }
```

133. WAP to input integers and store in a double dimension array of 3 rows and 4 columns. Finally, display the numbers in the form of a matrix. For example, the output should be as follows:

5	4	0	–1
3	2	1	6
2	1	9	8

Ans : COMP 2013

```
import java.util.*;
public class  {
     public static void main(String
args[]) {
        int a[][] = new int[3][4];
       System.out.println("Enter the
elements to the array");
      Scanner b = new Scanner(System.
in);
        for(int i = 0; i < 3; i++) {
              for(int j = 0; j < 4;
j++) {
               a[i][j] = b.nextInt();
            }
        }
        for(int i = 0; i < 3; i++) {
              for(int j = 0; j < 4;
j++) {
                System.out.print(a[i]
[j] + " ");
            }
            System.out.println();
        }
    }
}
```

Click Here to Buy book on Amazon

```
    Scanner obj = new Scanner(System.
in);
        int A[][] = new int[2][2];
        int B[][] = new int[2][2];
        int C[][] = new int[2][2];

           System.out.println("Enter
first matrix:");
       for (int i = 0; i < 2; i++) {
           System.out.println("Enter
the elements of " + i + " row");
            for (int j = 0; j < 2;
j++) {
                    A[i][j] = obj.
nextInt();
            }
        }

           System.out.println("Enter
second matrix:");
       for (int i = 0; i < 2; i++) {
           System.out.println("Enter
the numbers of " + i + " row");
            for (int j = 0; j < 2;
j++) {
                    B[i][j] = obj.
nextInt();
            }
        }

       System.out.println("Resulting
matrix:");
       for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2;
j++) {
               B[i][j]=obj.nextlnt;
}
}
              System.out.print(C[i]
[j] + " ");
for(int i=0;i<2;i+ +) {
for(int j=0;j<2;j++) {
         cIi]tjt =A[i]tjl +B[i][j];
System.out.print(c[i][j]+ " ");
            }
            System.out.println();
        }
    }
}
```

Download INDIA's Best Study app NODIA

139. Accept a matrix of 5 × 5 from the user and check whether it is an identity matrix or not. An identity matrix is one whose all diagonal elements are 1 and all other elements are zero.

Ans : COMP 2022

```
import java.util.*;
class {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int n[][] = new int[5][5];

       System.out.println("Enter the
matrix elements");
       for (int i = 0; i < 5; i++) {
          System.out.println("Enter
the no of " + i + "th row");
            for (int j = 0; j < 5;
j++) {
                    n[i][j] = obj.
nextInt();
            }
        }

        System.out.println();
        boolean v1 = false;
        boolean v2 = false;

       for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 5;
j++) {
              if (i == j) {
                 if (n[i][j] == 1)
                     v1 = true;
                } else if (n[i][j]
== 0) {
                    v2 = true;
              }
           }
        }

       if (v1 == true && v2 == true)
        System.out.println("Identity
matrix");
        else
            System.out.println("Not
identity");
    }
}
```

Downloaded from www.pdf.tube

CHAPTER 8

STRING HANDLING

NODIA

SUMMARY

1. STRING

A string is a sequence of characters used to represent text in Java. Strings are widely used in programming for tasks like displaying messages, reading input, manipulating text, and more. In Java, strings are objects of the String class, which is a part of the java.lang package.
Strings are immutable in Java, meaning once a string is created, its value cannot be changed. Any modification creates a new string object.

2. CREATING A STRING

1. Using the new Keyword : You can create a string object explicitly using the new keyword:
   `String s = new String("Muffin");`
   In the above statement, s is an object of class String which is initialized with the string literal "Muffin". It is equivalent to:
   `String s = "Muffin";`
2. Using String Literals : A simpler way to create strings is by assigning a string literal directly:
   `String s = "Muffin";`
   This method is more efficient because Java maintains a string pool. If a string with the same value already exists in the pool, it reuses the reference instead of creating a new object.

3. DIFFERENT WAYS TO INPUT A STRING

Java provides multiple ways to read strings from users or other sources. Here are some commonly used methods:

3.1 Passing a String as a Method Parameter

Strings can be passed to user-defined methods as parameters:

```
void check(String s) {
      System.out.println("The string is: " + s);
}
```

When the method is called:

```
check("Hello");
```

Output:

```
The string is: Hello.
```

3.2 Using the Scanner Class

The Scanner class from java.util is used for reading input from the console. To read a string:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
            System.out.print("Enter a word: ");
       String x = sc.next(); // Reads a single word (stops at whitespace)
              System.out.println("You entered: " + x);
    }
}
```

3.3 Using the BufferedReader Class

The BufferedReader class reads input efficiently, including entire lines of text. Its often used when performance is critical or for file I/O. Example:

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
public class Main {
    public static void main(String[] args) throws IOException {
                    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Enter a line: ");
        String y = br.readLine(); // Reads an entire line
              System.out.println("You
```

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

```
class {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
          int table[][] = new int[8]
[12];
        int i, j;
        for (i = 1; i < 8; i++) {
           for (j = 1; j < 12; j++) {
               table[i][j] = 1 / ((i
+ j) - 1);
                              System.out.
print(table[i][j]);
            }
            System.out.println();
        }
    }
}
```

143. Write a program to input integer elements into an array of size 20 and perform the following operations:

(i) Display the largest number from the array.

(ii) Display the smallest number from the array.

(iii) Display the sum of all the elements of the array.

Ans : MAIN 2017

```
public class {
    public static void main(String[]
args) {
        int a[] = new int[20];
     Scanner sc = new Scanner(System.
in);
        System.out.println("Enter 20
numbers:");
       for (int j = 0; j < 20; j++) {
            a[j] = sc.nextInt();
        }
        int largest = a[0];
        int smallest = a[0];
        int sum = 0;
       for (int j = 0; j < 20; j++) {
            if (a[j] > largest) {
                largest = a[j];
            }
            if (a[j] < smallest) {
                smallest = a[j];
            }
            sum += a[j];
        }
         System.out.println("Largest
number: " + largest);
         System.out.println("Smallest
number: " + smallest);
          System.out.println("Sum of
all elements: " + sum);
    }
}
```

144. Write a program to accept name and total marks of N number of students in two single subscript arrays name[ ] and totalmarks[ ].

(i) The average of the total marks obtained by N number of students.

$$\text{average} = \frac{\text{sum of total marks of all the students}}{\text{N}}$$

(ii) Deviation of each students total marks with the average.

$$\text{deviation} = \text{total marks of a student} - \text{average}$$

Ans : SQP 2018

```
import java.util.*;
class Marks {
    public static void main(int N) {
        double total = 0;
       String[] name = new String[N];
          double[] totalmarks = new
double[N];
       double[] dev = new double[N];
       System.out.println("Enter name
and total marks of each student:");
      Scanner sc = new Scanner(System.
in);
        for (int i = 0; i < N; i++) {
            name[i] = sc.next();
                    totalmarks[i] =
sc.nextDouble();
            total += totalmarks[i];
        }
        double avg = total / N;
        for (int j = 0; j < N; j++) {
              dev[j] = totalmarks[j]
- avg;
        }
           System.out.println("Name\
tTotalMarks\tDeviation");
        for (int k = 0; k < N; k++) {
          System.out.println(name[k]
+ "\t" + totalsmarks[k] + "\t" +
dev[k]);
        }
    }
}
}
```

www.nodia.in

Click Here to Buy book on Amazon

140. WAP in Java to accept the total sales done by three salesmen in four products. Use a double-dimension array to compute and print the total sales of each salesman and total sales of each product.

Ans : MAIN 2015

```
import java.util.*;
class  {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        int A[][] = new int[3][4];

        // Input for the sales array
        for (int i = 0; i < 3; i++) {
             for (int j = 0; j < 4;
j++) {
            System.out.println("Enter
the numbers:");
                    A[i][j] = obj.
nextInt();
            }
        }

        // Calculate total sales for
each salesman
        for (int i = 0; i < 3; i++) {
            int s = 0;
             for (int j = 0; j < 4;
j++) {
                s += A[i][j];
            }
            System.out.println("Total
sales by salesman " + (i + 1) + ": "
+ s);
        }

        // Calculate total sales for
each product
        for (int j = 0; j < 4; j++) {
            int s = 0;
             for (int i = 0; i < 3;
i++) {
                s1=s1 + A[i][j];
            }
int k=i+1
            System.out.println("Total
sales for product " + (j + 1) + ": "
+ s);
        }
    }
}
```

141. Using a double subscripted variable WAP in Java to enter data to form a 6 × 7 array and get a printout as follows:

SUN	MON	TUE	WED	THUR	FRI	SAT
–	–	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Ans : SQP 2018

```
import java.util.*;

class {
    public static void main(String[]
args) {
     Scanner obj = new Scanner(System.
in);
        String cal[][] = {
            {"SUN ", "MON ", "TUE ",
"WED ", "THUR", "FRI ", "SAT "},
            {"    ", "    ", " 1  ",
" 2  ", " 3  ", " 4  ", " 5  "},
            {" 6  ", " 7  ", " 8  ",
" 9  ", "10  ", "11  ", "12  "},
            {"13  ", "14  ", "15  ",
"16  ", "17  ", "18  ", "19  "},
            {"20  ", "21  ", "22  ",
"23  ", "24  ", "25  ", "26  "},
            {"27  ", "28  ", "29  ",
"30  ", "    ", "    ", "    "}
        };

        for (int i = 0; i < 6; i++) {
             for (int j = 0; j < 7;
j++) {
                System.out.print(cal[i]
[j]);
            }
            System.out.println();
        }
    }
}
```

142. Calculate elements of numeric table table which has 8 rows and 12 columns. Each element of table is calculated as:

$$\text{table}[i][j] = \frac{1}{i+j-1}$$

Ans : COMP 2019

```
import java.util.*;
```

NODIA

```
entered: " + y);
    }
}
```

4. COMPARISON OF INPUT METHODS

Method	Class	Use Case	Notes
check(String s)	None	Passing strings as arguments	Useful for passing predefined or dynamic values to methods.
Scanner.next()	Scanner	Reading single words from input	Simple and beginner-friendly; ignores spaces.
Scanner. nextLine()	Scanner	Reading full lines from input	Reads input until a newline character.
BufferedReader. readLine()	Buffered Reader	Reading entire lines of text	Efficient for large-scale input or files.

5. METHODS OF STRING CLASS

String class methods are listed below

trim()
toLowerCase()
toUpperCase()
length()
charAt(n)
indexOf()
lastIndexOf()
concat()
equals()
equalsIgnoreCase(s)
compareTo()
compareToIgnoreCase()
replace(c1, c2)
substring(i)
substring(i, j)
startsWith(s)
endsWith(s)
valueOf()

Download INDIA's Best Study app NODIA

6. TRIM()

This method of the String class returns a copy of the string after removing leading and trailing spaces.

Syntax

```
String variable = someString.trim();
```

Example:

```
String x = " Pan Cakes";
String y = " Pan Cakes".trim();
System.out.println(x);
System.out.println(y);
```

Output:

```
Pan Cakes
Pan Cakes (Spaces trimmed off from beginning & end)
```

7. TOLOWERCASE()

This method converts all the letters in the string to lowercase and returns the resulting string.

Syntax:

```
String variable = someString.toLowerCase();
```

Example:

```
String x = "Check this Out!".toLowerCase();
System.out.print(x);
```

Output:

```
check this out
```

8. TOUPPERCASE()

This method converts all letters of the string to uppercase and returns the resulting string.

Syntax:

```
String variable = someString.toUpperCase();
```

Example:

```
String x = "Check 2 this Out!".toUpperCase();
System.out.print(x);
```

Output:

```
CHECK 2 THIS OUT!
```

9. LENGTH()

This method returns an integer number that represents the total number of characters in the string.

Syntax:

```
int variable = someString.length();
```

Example:

```
int l = "string!! handling".length();
System.out.print(l);
```

Output:

```
16
```

Downloaded from www.pdf.tube

10. CHARAT()

This method returns the character at a specified index from the string.
Syntax:
```
char variable = someString.charAt(integer variable);
```
Example:
```
char t = "My country is my pride".charAt(7);
System.out.print(t);
```
Output:
t

11. INDEXOF()

This method returns an integer number that represents the index where the character/string exists in the main string.
```
int k = something.indexOf(character or string);
```
Example:
int k = "Country Side!".indexOf(!);
System.out.print(k);
Output:
12

12. LASTINDEXOF()

This method returns an integer number that represents the index of the last occurrence of the character/string in the main string.
Syntax:
```
int variable = someString.lastIndexOf(character or string);
```
Example:
```
int x = "Hello lulla".lastIndexOf("l");
System.out.print(x);
```
Output:
9

13. CONVERTING OTHER DATATYPES TO STRING TYPE

Using valueOf() method
Syntax:
```
String valueOf(long number);
String valueOf(int no);
```
General format:
```
String valueOf(datatypeVariableName);
```
Using toString() method
General format:
```
String variable = Wrapper class of datatype.toString(literal variable);
```
Example:
```
int a = 50;
String num = Integer.toString(a);
```

14. COMPARING STRINGS FOR EQUALITY

Syntax:
```
String s1, s2;
/* Assume that values are stored in s1 and s2 */
boolean result = s1.equals(s2);
/* If s1 and s2 are exactly the same, result is true; else result is false */
boolean res = s1.equalsIgnoreCase(s2);
/* If s1 and s2 store the same strings, which may differ in case (capital/small),
it gives true; else it gives false */
```

NODIA

15. COMPARING STRINGS FOR SORTING

Syntax:
```
String s1, s2;
/* Assume strings stored */
int d = s1.compareTo(s2);
```
(i) If the method returns d < 0, it means s1 is less than s2 in terms of its ASCII codes.
(ii) If d > 0, it means s1 is greater than s2.
(iii) If d $== 0$, it means s1 is the same as s2.

16. COMPARETOIGNORECASE()

This method compares two strings alphabetically while ignoring the case difference.
(i) If returned value $= 0$: Strings are the same.
(ii) If returned value < 0 : First string is less than the second string.
(iii) If returned value > 0 : First string is greater than the second string.
Syntax:
```
int variable = someString1.compareToIgnoreCase(someString2);
```
Example:
```
String s1 = "HELLO";
String s2 = "hello";
String s3 = "Hello";
int v1 = s1.compareToIgnoreCase(s3);
int v2 = s2.compareToIgnoreCase(s1);
System.out.print(v1 + "\t" + v2);
```
Output:
```
0    0
```

17. ASCII CODES

Every character on the keyboard has an associated integer code for its conversion to binary form.

This integer code is called the ASCII code.

(i) ASCII stands for American Standard Code for Information Interchange.

(ii) Range of ASCII codes:
  (i) From A to Z: 65 to 90
  (ii) From a to z: 97 to 122
  (iii) From 0 to 9: 48 to 57

18. REPLACE()

This method returns a new string that results after replacing all occurrences of the old character in the main string with the new character.
Syntax: String variable = somestring.replace(char old char, new char);
Example:

```
String n = "Hello".replace(l, m);
System.out.print(n);
```

Output: Hemmo

19. SUBSTRING(INDEX)

This method returns a new string which starts from the index till the end of the main string.
Syntax: String variable = somestring. substring(integer number or variable);
Example:

```
String p = "cut it out".substring(4);
System.out.print(p);
```

Output: it out

20. SUBSTRING(START INDEX, END INDEX)

This method returns a new string which starts from the start index and ends at end index 1.
Syntax: String variable = somestring. substring(integer, integer);
Example:

```
String p = "Cut it out".substring(0, 3);
System.out.print(p);
```

Output: Cut

21. STARTSWITH(S)

This method returns a boolean value true if the main string starts with the string argument; otherwise, it returns false.
Syntax: boolean variable = somestring. startsWith(string);
Example:

```
boolean v = "Cut".startsWith("e");
System.out.print(v);
```

Output: false

Download INDIA's Best Study app NODIA

22. ENDSWITH(S)

This method returns a boolean value true if the main string ends with the string argument; otherwise, it returns false.
Syntax: boolean variable = somestring. endsWith(string);
Example:

```
boolean v = "Cut".endsWith("t");
System.out.print(v);
```

Output: true

MULTIPLE CHOICE QUESTION

1. Write the output of the below code:
String t = "Reduce Reuse Recycle";
String x = new String(t);
System.out.print(x == t);

(a) True
(b) False
(c) Error
(d) 1

Ans : MAIN 2024

(b) False
The == operator compares references in Java. Here, t refers to a string literal, while x is a new object created using new String(t). Since x and t have different references in memory, the comparison x == t evaluates to false.

2. Write the output of the below code:
String t = "Reduce Reuse Recycle";
String x = new String(t);
System.out.print(x.equals(t));

(a) True
(b) False
(c) Error
(d) 1

Ans : SQP 2023

(a) true
The equals() method in Java compares the content of two strings, not their references. Here, both x and t have the same content: "Reduce Reuse Recycle". Thus, x. equals(t) evaluates to true.

3. Write the output of the below code:
String t = "Reduce Reuse Recycle";
String x = new String(t);
System.out.print(x.equalsIgnoreCase(t));

Click Here to Buy book on Amazon

(a) True
(b) False
(c) Error
(d) 1

Ans : COMP 2012

(a) true
The equalsIgnoreCase() method in Java compares the content of two strings while ignoring case differences. In this case, x and t both contain the same string "Reduce Reuse Recycle", with no case differences. Therefore, x.equalsIgnoreCase(t) evaluates to true.

4. The method compareTo() returns _____ if two strings are equal:
(a) True
(b) 1
(c) 0
(d) False

Ans : MAIN 2016

(c) 0
The compareTo() method in Java compares two strings lexicographically. If two strings are equal, their difference is zero, so the method returns 0. It does not return boolean values like true or false.

5. What is the output for the below statement?
System.out.print("Save electricity and water". indexOf(b) + 1);
(a) 3
(b) 0
(c) 1
(d) -1

Ans : SQP 2020

(b) 0
The indexof(char) method in Java returns the index of the first occurrence of the specified character in a string. If the character is not found, it returns -1. In this case, the character 'b' does not exist in the string "Save electricity and water". Therefore, indexof('b') returns -1, and adding 1 results in 0.

6. Give the output of the following string methods:
"MISSISSIPPI".indexof(S) + "MISSISSIPPI". lastIndexof(T)
(a) 10 (b) 12
(c) 20 (d) 11

Ans : COMP 2004

(b) 12
(i) The method indexof('S') returns the index of the first occurrence of 'S' in the string "MISSISSIPPI". The first 'S' occurs at index
(ii) The method lastIndexof('T') returns the index of the last occurrence of 'T' in the string. Since 'T' does not exist in "MISSISSIPPI", it returns -1.
The calculation becomes:
2+(-1)=

7. What is the output for:
System.out.print("Malayalam".replace(a, z).substring(3,6));
(a) Mzlzyzlzm
(b) Mzlyalam
(c) zyz
(d) zyzl

Ans : MAIN 2006

(c) zyz
replace('a', 'z'):
(i) The replace() method replaces all occurrences of 'a' with 'z' in the string "Malayalam", resulting in "Mzlyzlzym".
(ii) substring(3, 6):The substring() method extracts characters from index 3 (inclusive) to index 6 (exclusive).
For "Mzlyzlzym", the characters at indices 3, 4, and 5 are "yzl".
The final output is "yzl".

NODIA

The java class String is in _____ package:
(a) Java.String
(b) Java.lang
(c) Java.StringBuffer
(d) All of the above

Ans : SQP 2010

(b) Java.lang
The String class in Java is part of the java.lang package, which is automatically imported in every Java program. It does not belong to java.String or java.StringBuffer.

9. String s = new String(); will create an object of String with:
(a) No characters
(b) One character
(c) Space character
(d) None of the above

Ans : COMP 2013

(a) no characters

The statement String s = new String(); creates an empty String object, meaning it has no characters. The String will be initialized to an empty string (“”). It does not contain any characters, including spaces.

10. Which of the following expressions are invalid/ incorrect?

(a) String str = s1 s2;

(b) String str = s1 + s2;

(c) Boolean x = s1 <= s2;

(d) Boolean y = s1 == s2;

Ans : MAIN 2014

(a) and (c) are incorrect

(i) (a) String str = s1 s2; This is invalid because string concatenation in Java requires the + operator. Writing s1 s2 without an operator is a syntax error.

(ii) (b) String str = s1 + s2; This is valid. The + operator concatenates two strings in Java

(iii) (c) boolean x = s1 <= s2; This is invalid because relational operators like <= cannot be applied to strings in Java.

(iv) (d) boolean y = s1 == s2; his is valid, but it compares the references (memory locations) of the strings, not their content. To compare content, the .equals() method should be used.

11. Which of the following is the correct output of the below statements?
String str = new String();
System.out.print(“The new string is “ + str);

(a) The new string is

(b) The new string is null

(c) Exception

(d) The new string is str

Ans : SQP 2011

(a) The new string is

The statement String str = new String(); creates an empty string object, which is equivalent to “”. When concatenated with “The new string is “, the output will display “The new string is” followed by nothing, as the string is empty.

12. _ _ _ _ _ Class is used to convert a primitive data type to its corresponding object:

(a) String

(b) Wrapper

(c) System

(d) Math

Ans : COMP 2007

(b) Wrapper

The Wrapper class is used to convert a primitive data type (e.g., int, double, etc.) to its corresponding object (e.g., Integer, Double, etc.). These wrapper classes are part of the java. lang package and provide methods for converting primitives to objects and vice versa (autoboxing and unboxing).

13. Which package is the String class a part of in Java?

(a) java.util

(b) java.lang

(c) java.io

(d) java.text

Ans : MAIN 2008

(b) java.lang

The String class in Java is part of the java.lang package. This package is automatically imported into every Java program, so you don’t need to explicitly import it.

14. What is the default value of a String variable in Java if not initialized?

(a) null

(b) “”

(c) undefined

(d) “ “

Ans : SQP 2022

(a) null

In Java, if a String variable is declared as an instance variable and not explicitly initialized, its default value is null. This is because all reference types in Java have a default value of null.

15. What happens when you try to modify a String object in Java?

(a) The original string is modified.

(b) A new String object is created.

(c) An error occurs.

(d) The changes are ignored.

Ans : COMP 2015

(b) A new String object is created.

NODIA

Strings in Java are immutable, meaning their content cannot be changed once they are created. When you attempt to modify a String object, a new String object is created with the updated value, leaving the original string unchanged.

16. Which of the following is not a valid way to create a string in Java?

(a) String s = "Hello";
(b) String s = new String("Hello");
(c) String s = Hello;
(d) String s = new String();

Ans : MAIN 2018

(c) String s = Hello;

(i) (a) String s = "Hello";: This is valid. It creates a string literal and assigns it to s.

(ii) (b) String s = new String("Hello");: This is valid. It explicitly creates a new String object with the value "Hello".

(iii) (c) String s = Hello;: This is invalid because string literals in Java must be enclosed in double quotes ("").

(iv) (d) String s = new String();: This is valid. It creates an empty String object.

17. What does the equals() method in the String class compare?

(a) Memory location of strings
(b) Characters in the strings
(c) Case-insensitive equality of strings
(d) String length

Ans : SQP 2019

(b) Characters in the strings
The equals() method in the String class compares the content of the strings, specifically the sequence of characters, and determines whether they are equal. It is case-sensitive, meaning "Hello" is not equal to "hello".

18. Which method should you use to compare strings while ignoring case?

(a) Equals()
(b) CompareTo()
(c) EqualsIgnoreCase()
(d) CharAt()

Ans : COMP 2009

(c) EqualsIgnoreCase()
The equalsIgnoreCase() method in Java is used to compare two strings while ignoring case differences. It checks whether the content of the strings is the same, regardless of whether the characters are in uppercase or lowercase.

19. What is the output of the following code?

```
java
Copy code
String s = "hello";
System.out.println(s.toUpperCase());
```

(a) Hello (b) HELLO
(c) Hello (d) Error

Ans : MAIN 2003

(b) HELLO
The toUpperCase() method in Java converts all the characters of the string to uppercase. Here, the string s is "hello", and s.toUpperCase() will return "HELLO". The original string s remains unchanged because strings in Java are immutable.

20. What does the concat() method do in Java?

(a) Adds a character to the beginning of the string
(b) Combines two strings into one
(c) Splits a string into two parts
(d) Removes spaces from the string

Ans : SQP 2002

(b) Combines two strings into one
The concat() method in Java is used to combine (concatenate) two strings. It appends the specified string to the end of the original string and returns the new concatenated string. For example:
String s1 = "Hello";
String s2 = " World";
System.out.println(s1.concat(s2)); // Output: Hello World

21. Which method returns the number of characters in a string?

(a) Length()
(b) Size()
(c) Count()
(d) CharAt()

Ans : COMP 2001

(a) Length()
The length() method in Java returns the number of characters in a string, including spaces and special characters. It is a commonly used method to determine the string's length. For example:
String str = "Hello";
System.out.println(str.length()); // Output: 5

Downloaded from www.pdf.tube

22. What is the result of the following code?

```
String s = "  Hello World  ";
System.out.println(s.trim());
```

(a) " Hello World"
(b) "Hello World "
(c) "Hello World"
(d) "HelloWorld"

Ans : MAIN 2005

(c) "Hello World"
The trim() method in Java removes any leading and trailing whitespace from the string but does not affect spaces between words. In this case, the string " Hello World " becomes "Hello World" after applying trim().

23. Which method is used to extract a portion of a string?
(a) substring()
(b) indexOf()
(c) replace()
(d) split()

Ans : SQP 2017

(a) substring()
The substring() method in Java is used to extract a portion of a string. It takes either one parameter (starting index) or two parameters (starting and ending indices) and returns the substring between those indices. For example:
String s = "Hello World";
System.out.println(s.substring(0, 5)); // Output: Hello

24. What is the result of indexOf("a") for the string "banana"?
(a) 0 b) 1
(c) 2 d) 3

Ans : COMP 2021

(b) 1
The indexOf("a") method in Java returns the index of the first occurrence of the character "a" in the string "banana". In this case:
The first "a" appears at index 1 (0-based indexing).
So, the result is 1.

25. What does the lastIndexof() method do?
(a) Finds the first occurrence of a character
(b) Finds the last occurrence of a character
(c) Finds the middle index of a string
(d) Finds the length of a string

Ans : MAIN 2024

(b) Finds the last occurrence of a character
The lastIndexof() method in Java is used to find the last occurrence of a specified character or substring within a string. It returns the index of the last occurrence, or -1 if the character or substring is not found.
String s = "banana";
System.out.println(s.lastIndexOf('a')); // Output: 5

NODIA

26. Which method replaces all occurrences of a character in a string?
(a) replace()
(b) replaceAll()
(c) substring()
(d) charAt()

Ans : SQP 2023

(a) replace()
The replace() method in Java is used to replace all occurrences of a specific character or substring in a string with another character or substring. For example:
String s = "banana";
System.out.println(s.replace('a', 'o')); // Output: bonono

27. What is the result of compareTo("apple") for the string "banana"?
(a) A positive number
(b) A negative number
(c) 0
(d) null

Ans : COMP 2012

(a) A positive number

The compareTo() method in Java compares two strings lexicographically. It returns:

(i) A positive number if the calling string is lexicographically greater than the argument.

(ii) A negative number if the calling string is lexicographically smaller than the argument.

(iii) 0 if both strings are equal.

Here, "banana".compareTo("apple") compares the strings character by character. Since 'b' comes after 'a' lexicographically, the result will be a positive number.

Download INDIA's Best Study app NODIA

28. What does the startsWith("J") method return for the string "Java"?
(a) True
(b) False
(c) "Java"
(d) An error

Ans : MAIN 2016

(a) True
The startsWith() method in Java checks if the string begins with the specified prefix. For the string "Java", the method startsWith("J") checks whether the string starts with the character 'J'. Since "Java" does start with 'J', the method returns true.

29. What does the value of() method do?
(a) Converts a string to uppercase
(b) Converts a string to a number
(c) Converts other data types to a string
(d) Splits a string into an array

Ans : SQP 2020

(c) Converts other data types to a string
The value of() method in Java converts other data types (such as integers, floats, booleans, characters, or objects) into their string representation. For example:
int number = 42;
String str = String.valueof(number);
System.out.println(str); // Output: "42"

30. Which method checks if two strings are identical?
(a) ==
(b) equals()
(c) compareTo()
(d) contains()

Ans : COMP 2004

(b) equals()
The equals() method in Java checks if two strings are identical by comparing their content (character by character). It returns true if the strings are the same and false otherwise.
For example:
String s1 = "hello";
String s2 = "hello";
System.out.println(s1.equals(s2));
// Output: true

Download INDIA's Best Study app NODIA

31. What does the toLowerCase() method do?
(a) Converts the string to lowercase
(b) Converts the string to uppercase
(c) Removes spaces from the string
(d) Splits the string into an array

Ans : MAIN 2006

(a) Converts the string to lowercase
The toLowerCase() method in Java converts all the characters in a string to lowercase. It does not affect spaces, special characters, or the structure of the string. For example:
String s = "Hello World";
System.out.println(s.toLowerCase());
// Output: hello world

32. Which method checks if a string ends with a specific character or substring?
(a) endsWith()
(b) startsWith()
(c) contains()
(d) substring()

Ans : SQP 2010

(a) endsWith()
The endsWith() method in Java checks whether a string ends with a specified character or substring. It returns true if the string ends with the specified suffix and false otherwise.
For example:
String s = "Hello World";
System.out.println(s.endsWith("World"));
// Output: true

33. A programmer wants to check if a string str starts with the letter "J". They write the following incorrect code:
`boolean result = str.startWith("J");`
What will be the correct statement?
A. Boolean result = str.startsWith("J");
B. Boolean result = str.startsWith(J);
C. Boolean result = startsWith(str, "J");
(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans : COMP 2013

(a) Only A
A. Boolean result = str.startsWith("J");: This is correct. The startsWith() method checks if the string begins with the specified prefix.
B. Boolean result = str.startsWith(J);: This is

Click Here to Buy book on Amazon

incorrect because "J" must be enclosed in double quotes to denote a string.

C. boolean result = startsWith(str, "J");: This is incorrect because startsWith is not a standalone function in Java; it must be called on a string object like str.startsWith().

34. A student writes the following statement to calculate the length of a string str with value "Java".

```
int len = str.size();
```

Which of the following is correct?

A. int len = str.length();
B. int len = length(str);
C. int len = str.charAt();

(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans : MAIN 2014

(a) Only A

The size() method does not exist for strings in Java. The correct way to calculate the length of a string is by using the .length() method.

A. int len = str.length();: This is correct. The length() method returns the number of characters in the string.

B. int len = length(str);: This is incorrect because there is no such standalone length() function for strings in Java.

C. int len = str.charAt();: This is incorrect because charAt() is used to retrieve a character at a specific index, not the length of a string.

35. A developer wants to concatenate two strings str1 and str2. They write the incorrect code:

String result = str1.concat(str2);

Which of the following will work correctly?

A. String result = str1.concat(str2);
B. String result = str1 + str2;
C. String result = str1.add(str2);

(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans : SQP 2011

(c) Both A and B

A. String result = str1.concat(str2);:This is correct. The concat() method appends str2 to the end of str1 and returns the concatenated string.

B. String result = str1 + str2;: This is also correct. The + operator concatenates two strings in Java.

C. String result = str1.add(str2);: This is incorrect because the add() method does not exist for strings in Java.

36. A student writes the following code to replace all occurrences of a with o in the string "Java":

```
String result = str.replace("a", "o");
```

What will be the correct statement?

A. String result = str.replace(a, o);
B. String result = str.replace("a", "o");
C. String result = str.change(a, o);

(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans : COMP 2007

(c) Both A and B

A. String result = str1.concat(str2);: This is correct. The concat() method appends str2 to the end of str1 and returns the concatenated string.

B. String result = str1 + str2;: This is also correct. The + operator concatenates two strings in Java.

C. String result = str1.add(str2);: This is incorrect because the add() method does not exist for strings in Java.

37. A programmer wants to extract the substring "Java" from the string "I love Java programming". They write the incorrect code:

```
String sub = str.substr(7, 11);
```

What will be the correct statement?

A. String sub = str.substring(7, 11);
B. String sub = str.substring(7);
C. String sub = substring(str, 7, 11);

(a) Only A
(b) Only B
(c) Both A and B
(d) All the three

Ans : MAIN 2008

(a) Only A

A. String sub = str.substring(7, 11);: This is correct. The substring(start, end) method extracts the substring starting at index 7 (inclusive) and ending at index 11 (exclusive), which gives "Java".

NODIA

(ii) Differentiate between to LowerCase () and to UpperCase () methods

Ans : MAIN 2013

(i) (a) Helloworld
(b) 5
(c) 1
(d) false

(ii) toLowerCase method converts all the alphabets of a string to lowercase, whereas toUpperCase method converts all the alphabets of a string to uppercase.

Example:

```
String x = "Hello";
String y = x.toLowerCase();
String z = x.toUpperCase();
System.out.print(y + " " + z);
```

Output: hello HELLO

62. What will the following code output?

```
String s = "malayalam";
System.out.println(s.indexOf(m));
System.out.println(s.lastIndexOf(m));
```

Ans : SQP 2014

(i) 0
(ii) 8

63. What is the data type returned by the library functions:

(i) compareTo()
(ii) equals()

Ans : MAIN 2007

(i) compareTo() Returns result of integer data type.
(ii) equals() Returns result of boolean data type.

64. How does Java implement a string?

Ans : SQP 2008

Java implements a string as a contiguous sequence of characters. A string variable is implemented as an object of class String.

65. State the output when the following program segment is executed:

```
String a = "Smartphone", b = "Graphic Art";
String h = a.substring(2, 5);
String k = b.substring(8).toUpperCase();
System.out.println(h);
System.out.println(k.equalsIgnoreCase(h));
```

Ans : COMP 2022

Here, h = art and k = ART
Output:
art
true

66. What is the difference in the classes String and String Buffer?

Ans : MAIN 2015

The objects of class String are of fixed length. Hence, insertion of characters or substrings in the middle or at the end of that string object is not possible. The String Buffer class creates strings whose length can be changed. Hence, the modifications are possible in the same string object.

67. State the output of the following program segment:

```
String str1 = "great";
String str2 = "minds";
System.out.println(str1.
substring(0,2).concat(str2.
substring(1)));
System.out.println("WH"    +    str1.
substring(2).toUpperCase());
```

Ans : COMP 2011

Output
grinds
WHEAT

68. In which package are the two classes String and String Buffer defined?

Ans : SQP 2018

Java.lang package.

69. Explain briefly how Java manages strings?

Ans : COMP 2024

Java handles its strings using two classes String and String Buffer. Strings are created as objects of these classes. They are not implemented as an array of characters.
Implementation of strings as objects makes string handling easier and more manageable in Java as operations are applied to strings rather than arrays of characters.
All other data types can be converted to strings, and even string numbers can be converted to numeric data types.
Many manipulation tasks can be applied to strings using methods provided in classes String and String Buffer.

Downloaded from www.pdf.tube

(iii) Print the value of subStr using System.out. println() ((3)).

42. Consider the following program segment to check if a string starts with a specific character. The statements are jumbled. Choose the correct order of statements.

```
void checkStart(String str) {
boolean starts;                    // (1)
starts = str.startsWith("A");// (2)
System.out.println(starts);// (3)
}
```

(a) (1) (2) (3)
(b) (1) (3) (2)
(c) (2) (1) (3)
(d) (3) (1) (2)

Ans : COMP 2009

(a) (1) (2) (3)
To check if a string starts with a specific character and print the result, the statements should follow this logical sequence:

(i) Declare the variable starts to store the result of the check ((1)).
(ii) Use the str.startsWith("A") method to check if the string starts with the character 'A' and assign the result to starts ((2)).
(iii) Print the value of starts using System.out. println() ((3)).

43. **Assertion (A) :** Strings in Java are immutable.
Reason (R) : Any modification to a string creates a new string object instead of modifying the original string.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2003

(a) Both A and R are true, and R is the correct explanation of A.
In Java, strings are immutable, meaning their content cannot be changed after they are created. Any operation that appears to modify a string (e.g., concatenation, replacement) actually creates a new string object. This behavior explains why strings in Java are immutable.

Download INDIA's Best Study app NODIA

44. **Assertion (A) :** The length() method in the String class is used to determine the number of characters in a string.
Reason (R) : The length() method includes spaces and special characters in its count.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : SQP 2002

(a) Both A and R are true, and R is the correct explanation of A.
The length() method in the String class returns the total number of characters in a string, including spaces and special characters. This means the Reason (R) correctly explains how the length() method works. Therefore, both the Assertion (A) and the Reason (R) are true, and R is the correct explanation of A.

45. **Assertion (A) :** The to Uppercase() method in the String class converts all the characters of a string to uppercase.
Reason (R) : The to Uppercase() method modifies the original string.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : COMP 2001

(c) A is true, but R is false.
The to Uppercase() method in the String class converts all the characters of a string to uppercase, so Assertion (A) is true. However, strings in Java are immutable, so the to Uppercase() method does not modify the original string but instead returns a new string with the characters converted to uppercase.

NODIA

46. **Assertion (A) :** The substring() method extracts a portion of a string based on the specified start and end indices.
Reason (R) : The substring() method modifies the original string by removing characters outside the specified range.

(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : MAIN 2005

(c) A is true, but R is false.
The substring() method in Java extracts a portion of a string based on the specified start (inclusive) and end (exclusive) indices, so Assertion (A) is true. However, strings in Java are immutable, and the substring() method does not modify the original string but instead returns a new string.

47. **Assertion (A) :** The replace() method in the String class replaces all occurrences of a specified character in the string with another character.
Reason (R) : The replace() method creates a new string with the replacements and does not modify the original string.
(a) Both A and R are true, and R is the correct explanation of A.
(b) Both A and R are true, but R is not the correct explanation of A.
(c) A is true, but R is false.
(d) A is false, but R is true.

Ans : SQP 2017

(a) Both A and R are true, and R is the correct explanation of A.
The replace() method in the String class replaces all occurrences of a specified character (or substring) in the original string with another character (or substring), so Assertion (A) is true. Since strings in Java are immutable, the replace() method creates a new string with the replacements and does not modify the original string, making Reason (R) true and correctly explaining A.

TWO MARKS QUESTIONS

48. Write a statement each to perform the following tasks on a string:
(i) Find and display the position of the last space in a string s
(ii) Convert a number stored in a string variable x to double data type.

Ans : SQP 2024

(i) System.out.println(s.lastIndexOf( ) + 1); // +1 gives the position
(ii) double d = Double.parseDouble(x);

49. Give the output of the following:
String n = "Computer Knowledge.";
String m = "Computer Applications";
System.out.println(n.substring(0,8).concat(m.substring(9)));
System.out.println(n.endsWith("e"));

Ans : COMP 2024

ComputerApplications
true

50. Write the output of the following:
(i) System.out.println(Character.isUpperCase(R));
(ii) System.out.println(Character.toUpperCase(J));

Ans : MAIN 2021

(i) true
(ii) J

51. State the purpose and return data type of the following String functions:
(i) indexOf()
(ii) compareTo()

Ans : COMP 2023

(i) It returns the index of the first occurrence of a character in a string.
(ii) It compares two strings and returns 0 if they are the same. -1 if the first string is less than the second string. 1 if the first string is greater than the second string.

52. Write the return type of the following library functions:
replace(char, char)

Ans : MAIN 2016

String

53. Write a statement each to perform the following task on a string:
(i) Extract the second last character of a word stored in the variable wd.
(ii) Check if the second character of a string str is in uppercase.

Ans : SQP 2012

(i) char x = wd.charAt(wd.length() 2);

Click Here to Buy book on Amazon

(ii) boolean y = Character.isUpperCase(str.charAt(1));

54. (i) State the output of the following program segment:
String s = "Examination";
int n = s.length();
System.out.println(s.startsWith(s.substring(5, n)));
System.out.println(s.charAt(2) == s.charAt(6)));
(ii) System.out.println(x[3]);

Ans : COMP 2019

(i) false (b) true
(ii) Big data

55. If String x = "Computer"; String y = "Applications";
what do the following functions return for:
(i) System.out.println(x.substring(1, 5));
(ii) System.out.println(x.indexOf(x.charAt(4)));
(iii) System.out.println(y + x.substring(5));
(iv) System.out.println(x.equals(y));

Ans : MAIN 2020

(i) omput
(ii) 4
(iii) Applicationster
(iv) False

56. Write statements to show how finding the length of a character array char[] differs from finding the length of a String object str

Ans : SQP 2013

The program code is as follows:

```
char arr[] = new char[10];
int len = arr.length;
System.out.print("Length of the character array is: " + len);
String str = "Embrace your blessings";
int l = str.length();  // Note the parenthesis () in finding the length of the string
System.out.print("Length of the string is: " + l);
```

57. What is the output of the following:
(i) System.out.println("four :" + 4 + 2);
System.out.println("four :" + (2 + 2));
(ii) String S1 = "HI";
String S2 = "HI";
String S3 = "there";
String S4 = "hi";
System.out.println(S1 + "equals" + S2 + "→" + S1.equals(S2));
System.out.println(S1 + "equals" + S3 + "→" + S1.equals(S3));
System.out.println(S1 + "equals" + S4 + "→" + S1.equals(S4));
System.out.println(S1 + "(EqualsIgnoreCase " + S4 + "→" + S1.equalsIgnoreCase(S4));

Ans : COMP 2004

(i) four :42
four :4
(ii) HI equals HI → true
HI equals there → false
HI equals hi → false
HI EqualsIgnoreCase hi → true

58. State the data type and values of a and b after the following segment is executed:
String s1 = "Computer", s2 = "Application";
a = (s1.compareTo(s2));
b = (s1.equals(s2));

Ans : MAIN 2012

Data type of a is int
Data type of b is Boolean
a = 1
b = false

59. What will be the output for the following program segment:

```
String s = new String("abc");
System.out.println(s.toUpperCase());
```

Ans : SQP 2006

Output : ABC

60. State the values stored in the variables str1 and str2:
String s1 = "good";
String s2 = "world matters";
String str1 = s2.substring(5).replace(t, n);
String str2 = s1.concat(str1);

Ans : COMP 2010

str1 = "manners"
str2 = "good manners"

61. (i) What do the following functions return for:
String x = "hello";
String y = "world";
(a) System.out.println(x + y);
(b) System.out.println(x.length());
(c) System.out.println(x.charAt(3));
(d) System.out.println(x.equals(y));

Downloaded from www.pdf.tube

(u)
```
if (name.startsWith("Z")) {
    System.out.println("Hello Boss");
} else if (name.endsWith("a")) {
        System.out.println("Good Day Boss");
```
(v)
```
String numberStr = String.valueOf(56.78);
```
(w)
```
if (address.endsWith("34")) {
    System.out.println("yahooooo!");
} else {
    System.out.println("Im sad");
}
```

72. Write the output for the following:
String s = "Today is Test";
System.out.println(s.indexOf(T));
System.out.println(s.substring(0, 7) + " " + "Holiday");

Ans : COMP 2017

(i) 0
(ii) Today is Holiday

73. Give the output of the following code:
String A = "26", B = "100";
String D = A + B + "200";
int x = Integer.parseInt(A);
int y = Integer.parseInt(B);
int d = x + y;
System.out.println("Result 1 = " + D);
System.out.println("Result 2 = " + d);

Ans : MAIN 2017

Result 1 = 26100200
Result 2 = 126

74. Consider the following code and predict the output:
String X = "50", Y = "25";
String Z = X + Y + "75";
int a = Integer.parseInt(X);
int b = Integer.parseInt(Y);
int sum = a + b + 75;
System.out.println("Concatenation Result = " + Z);
System.out.println("Arithmetic Result = " + sum);

Ans : SQP 2019, 2018

Concatenation Result = 502575
Arithmetic Result = 150

75. Write the return data type of the following function:
(i) endsWith()
(ii) startsWith()
(iii) random()

Ans : COMP 2016

(i) boolean
(ii) boolean
(iii) double

76. Give the output of the following string functions:
(i) "ACHIEVEMENT".replace(E, A)
(ii) "DEDICATE".compareTo("DEVOTE")

Ans : MAIN 2018

(i) ACHIAVAMANT
(ii) 1

FIFTEEN MARKS QUESTIONS

77. Write a program to accept a word and convert it into lowercase if it is in uppercase, and display the new word by replacing only the vowels with the character following it.
Example:
Sample Input: computer
Sample Output: cpmpvtfr

Ans : SQP 2024

```
import java.util.*;
public class ReplaceVowel {
    public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
         System.out.println("Enter a string");
        String str = sc.next();
            String strLower = str.toLowerCase();
        String newstr = "";

       for (int i = 0; i < strLower.length(); i++) {
                char c = strLower.charAt(i);
            switch(c) {
}
case A:
case e:
case E:
```

NODIA

70. Write the two ways in which you can assign a text string "TOGETHER" to a string variable, say, str.

Ans : MAIN 2023

(i) String str = "TOGETHER";

(ii) char arr[] = {T, O, G, E, T, H, E, R};
String str = new String(arr);

71. Write the Java statements to perform the following tasks on strings :

(a) Initialize a variable city to store "DELHI".

(b) Accept the address from the user.

(c) Print the number of characters in the above-entered address.

(d) Concatenate city with address and store it as string Postal Address.

(e) Extract 10 characters from the 1st character to the 10th of Postal Address.

(f) Extract the last character of Postal Address.

(g) Compare city with "Ghaziabad" and store the boolean result in the variable bool.

(h) Print the position of the first occurrence of the letter B in string City.

(i) Print the position of the last occurrence of the substring "DEL" in the string str1.

(j) Replace all occurrences of the letter m by M in the text "my mother is very loving".

(k) Compare name with "KARAM ARORA", ignoring case matching. Store the boolean result in bln.

(l) Compare the ASCII codes of two string variables str1 and str2, and store the result in the variable comparison.

(m) To Search for a substringsrch_str in another string orig_str and assign the returned position to pos.

(n) To Search for a substring srch_str starting from the 10th position in orig_str and assign the returned position to pos1.

(o) Extract a substring from string str from the 4th position to the 10th position and assign it to str2.

(p) Replace all occurrences of character ch1 by ch2 in the string str.

(q) Reverse the string "SUN MICROSYSTEMS".

(r) Remove all leading and trailing white spaces from a string stored in the variable stng.

(s) Change the case of all characters of a string str to lowercase.

(t) Change the case of all characters of a string str2 to uppercase.

(u) Print "Hello Boss" if a given name starts with the letter Z, else if the names last letter is a, print "Good Day Boss".}

(v) Convert the number 56.78 to its equivalent string.

(w) If the pin code of the address ends with 34, display "yahoo!", Otherwise display "Im sad"

Ans : SQP 2012

(a) `String city = "DELHI";`

(b)
```
BufferedReaderob=new
BufferedReader(new
InputStreamReader (System.in));
System.out.println("Please Enter
your address:");
String address = ob.readline( );
```

(c) `System.out.println(address.length( ) + "is the number of characters in the address");`

(d) `String PostalAddress = city.concat(address);`

(e) `String sub = PostalAddress.substring(0, 10);`

(f) `char lastChar = PostalAddress.charAt(PostalAddress.length() 1);`

(g) `boolean bool = city.equals("Ghaziabad");`

(h) `int position = city.indexOf(B);`

(i) `int lastPos = str1.lastIndexOf("DEL");`

(j) `String replacedText = "my mother is very loving".replace(m, M);`

(k) `boolean bln = name.equalsIgnoreCase("KARAM ARORA");`

(l) `int comparison = str1.compareTo(str2);`

(m) `int pos = orig_str.indexOf(srch_str);`

(n) `int pos1 = orig_str.indexOf(srch_str, 10);`

(o) `String str2 = str.substring(4, 11);`

(p) `String replacedStr = str.replace(ch1, ch2);`

(q)
```
String reversed = new
StringBuilder("SUN
MICROSYSTEMS").reverse().
toString();
```

(r) `String trimmed = stng.trim();`

(s) `String lower = str.toLowerCase();`

(t) `String upper = str2.toUpperCase();`

Click Here to Buy book on Amazon

B. String sub = str.substring(7);: This is incorrect because substring(start) extracts the substring from the start index to the end of the string, which would result in "Java programming".

C. String sub = substring(str, 7, 11);: This is incorrect because substring() is not a standalone function in Java; it must be called on a string object.

38. Consider the following program segment to find the length of a string. The statements are jumbled. Choose the correct order of statements.

```
void findLength(String str) {
    int length;           // (1)
   System.out.println(length);// (2)
    length = str.length();  // (3)
}
```

(a) (1) (3) (2)
(b) (3) (1) (2)
(c) (2) (1) (3)
(d) (1) (2) (3)

Ans : SQP 2022

(a) (1) (3) (2)
To find the length of a string and print it, the statements must be ordered logically:

(i) Declare the variable length to hold the length of the string ((1)).

(ii) Assign the length of the string using str. length() ((3)).

(iii) Print the value of the variable length ((2)).

39. Consider the following program segment to convert a string to uppercase. The statements are jumbled. Choose the correct order of statements.

```
void toUpperCase(String str) {
   String upperStr;          // (1)
 upperStr = str.toUpperCase();// (2)
System.out.println(upperStr); // (3)
}
```

(a) (1) (2) (3)
(b) (2) (1) (3)
(c) (1) (3) (2)
(d) (3) (1) (2)

Ans : COMP 2015

(a) (1) (2) (3)
To find the length of a string and print it, the statements must be ordered logically:

(i) Declare the variable length to hold the length of the string ((1)).

(ii) Assign the length of the string using str. length() ((3)).

(iii) Print the value of the variable length ((2)).

40. Consider the following program segment to replace characters in a string. The statements are jumbled. Choose the correct order of statements.

```
void replaceCharacters(String str) {
String replacedStr;          // (1)
replacedStr = str.replace(a, o);  //
(2)
System.out.println(replacedStr);//
                                   (3)
}
```

(a) (1) (3) (2)
(b) (1) (2) (3)
(c) (2) (1) (3)
(d) (3) (2) (1)

Ans : MAIN 2018

(b) (1) (2) (3)
To replace characters in a string and print the result, the statements should be ordered logically:

(i) Declare the variable replacedStr to store the modified string ((1)).

(ii) Replace characters in the string using str. replace('a', 'o') and assign the result to replacedStr ((2)).

(iii) Print the value of replacedStr ((3)).

41. Consider the following program segment to extract a substring from a string. The statements are jumbled. Choose the correct order of statements.

```
void extractSubstring(String str) {
    String subStr;              // (1)
subStr = str.substring(5, 10);// (2)
 System.out.println(subStr);// (3)
}
```

(a) (1) (3) (2)
(b) (2) (1) (3)
(c) (1) (2) (3)
(d) (3) (2) (1)

Ans : SQP 2019

(c) (1) (2) (3)
To extract a substring and print it, the statements should follow this logical sequence:

(i) Declare the variable subStr to store the extracted substring ((1)).

(ii) Extract the substring using str.substring(5, 10) and assign it to subStr ((2)).

NODIA

```
case i:
case I:
case o:
case O:
case u:
case U:
    newstr = newstr + (char) (c + 1);
    break;
default:
    newstr = newstr + c;
}
}
System.out.println("New string is: "
+ newstr);
}
```

Variables Used :
str: Inputted string.
strLower: String converted to lowercase.
newstr: Stores the final new value of the string.
i: Loop variable.

78. Write a program to accept a string, convert the string to uppercase, count, and output the number of double-letter sequences that exist in the string.
Sample Input:
"SHE WAS FEEDING THE LITTLE RABBIT WITH AN APPLE"
Sample Output: 4

Ans : COMP 2020

```
package sample;
import java.util.Scanner;
public class Ans2 {
    public static void main(String[]
args) {
    Scanner scan = new Scanner(System.
in);
          System.out.println("Please
enter a string: ");
       String str = scan.nextLine();
        str = str.toUpperCase();

        int len = str.length();
        char c, d;
        int count = 0;

         for (int i = 0; i < len 1;
i++) {
            c = str.charAt(i);
            d = str.charAt(i + 1);
            if (c == d) {
                count++;
            }
        }

       System.out.println("Number of
double-letter sequences: " + count);
    }
}
```

79. Write a program to assign a full path and file name as given below. Using library functions, extract and output the file path, file name, and file extension separately as shown.
Input: C:\Users\admin\Pictures\flower.jpg
Output: Path: C:\Users\admin\Pictures\
File name : flower
Extension : jpg

Ans : MAIN 2014

```
import java.util.Scanner;
public class FileDetails {
    public static void main(String[]
args) {
    Scanner scan = new Scanner(System.
in);
       System.out.println("Enter the
full file name: ");
        String file = scan.next();

            int lastIndex = file.
lastIndexOf("\\");
           String filePath = file.
substring(0, lastIndex);
             String name = file.
substring(lastIndex + 1);
          String filename = name.
substring(0, name.indexOf("."));
         String extension = name.
substring(name.indexOf(".") + 1);

          System.out.println("File
Path: " + filePath);
        System.out.println("Name: "
+ filename);
      System.out.println("Extension:
" + extension);
    }
}
```

Variables Used :
file : Full file name input by the user.
lastIndex : Last index of the character \ in the full file name.
name : File name including the extension.
filename, extension : Stores the name of the file and extension, respectively.

Downloaded from www.pdf.tube

88. Write a complete Java program to concatenate two strings str1 and str2 entered from the keyboard.

Ans : MAIN 2022

```
import java.util.*;
public class Concat {
     public static void main(String
args[]) {
      Scanner sc = new Scanner(System.
in);
          System.out.println("Please
enter two strings:");
        String str1 = sc.next();
        String str2 = sc.next();
     String str3 = str1.concat(str2);
              System.out.print("The
concatenated string is: " + str3);
    }
}
```

89. Define a method replace Substring(String input) to replace the first occurrence of the word "apple" with "orange". The modified string should be returned to the main method, where it is displayed.

Ans : SQP 2024

```
public class Replace Substring {
     // Define the method to replace
the first occurrence of "apple" with
"orange"
            public   static   String
replaceSubstring(String input) {
       // Replace the first occurrence
of "apple" with "orange"
       return input.replaceFirst("\\
bapple\\b", "orange");
    }
public  static  void  main(String[]
args) {
        // Example input string
         String input = "An apple a
day keeps the doctor away. The apple
is sweet.";
// Call the replaceSubstring method
             String modifiedString =
replaceSubstring(input);
// Display the original and modified
strings
        System.out.println("Original
String: " + input);
         System.out.println("Modified
String: " + modifiedString);
    }
}
```

90. Define a method replace Word(String input) to replace the last occurrence of the word "is" with "was". The modified string should be returned to the main method, where it is displayed.

Ans : COMP 2015

```
public class ReplaceLastOccurrence {

   // Define the method to replace the
last occurrence of "is" with "was"
            public   static   String
replaceWord(String input) {

         // Find the last occurrence
of the word "is"
            int lastIndex = input.
lastIndexOf(" is ");
        if (lastIndex == -1) {
          return input; // Return the
original string if "is" is not found
        }

       // Replace the last occurrence
            String before = input.
substring(0, lastIndex);
             String after = input.
substring(lastIndex + 4);
          return before + " was " +
after;
    }

    public static void main(String[]
args) {
        // Example input string
          String input = "This is a
test. It is just an example.";

       // Call the replaceWord method
             String modifiedString =
replaceWord(input);

         // Display the original and
modified strings
        System.out.println("Original
String: " + input);
         System.out.println("Modified
String: " + modifiedString);
    }
}
```

91. Define a method replaceString to replace the first occurrence of the word "an" with "the".

Ans : MAIN 2018

NODIA

```
     String str1 = str.toUpperCase();
      System.out.println("Characters
" + "\t" + "Frequency");
        int l = str.length();
       System.out.println("Length of
string is: " + l);

         for (char ch = A; ch <= Z;
ch++) {
            int n = 0;
             for (int j = 0; j < l;
j++) {
                  char ch1 = str1.
charAt(j);
               if (ch1 == ch) {
                   n++;
               }
            }
            if (n != 0) {
               System.out.println(ch
+ "\t" + n);
            }
        }
    }
}
```

82. Write a program to input a sentence and print the number of characters found in the longest word of the given sentence.
Example:
If S = "India is my country" , the output should be 7

Ans : MAIN 2010

```
class Longest {
     public static void main(String
S) {
          // String S inputted as a
parameter of the function.
       String[] words = S.split(" ");
        int maxLength = 0;
        for (String word : words) {
                if (word.length() >
maxLength) {
                   maxLength = word.
length();
            }
        }
             System.out.println("The
longest word has " + maxLength + "
characters.");
    }
}
```

```
int k = 0;
int l = S.length();
int i, j = 0;
String str;
int len[] = new int[ ];

for (i = 0; i <= l 1; i++) {
    char c = S.charAt(i);

    // Assume words are separated by
a single space
    if (c ==  ) {
        // Extract the word
        str = S.substring(j, i);
        len[k] = str.length();
        k++;
        j = i + 1;
    }
}

int max = 0;
for (int x = 0; x <= k 1; x++) {
    if (len[x] > max)
        max = len[x];
}
System.out.print("Length    of    the
longest word is " + max);
}
```

Variables Used :
S: Inputted string.
l: Length of the inputted string.
len: Array variable that stores the length of every word in the string.
str: Stores each word of a string.
k: Subscript for array len.
i, x: Loop variables.
j: Index from where the word starts in the string S. Assume the first character is not a space.
max: Variable to store the maximum length.

83. Write a program to input a string and print out the text with the uppercase and lowercase letters reversed, but all other characters should remain the same as before.
Example:
INPUT: WelComE TO School
OUTPUT: WELcOMe to sCHOOL

Ans : SQP 2013

```
class Changecase {
     public static void main(String
str) {
        int i, l = str.length();
        String s = "";
```

Click Here to Buy book on Amazon

80. Write a program to input a string in lowercase and print the frequency of vowels in the string.
Example:
INPUT: programming is fun
OUTPUT:

Vowel	Frequency
a	1
e	0
i	2
o	1
u	1

Ans : SQP 2004

```
import java.util.Scanner;

public class VowelFrequency {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Input the string
           System.out.print("Enter a
string in lowercase: ");
       String input = sc.nextLine();

          // Initialize an array to
store frequency of vowels
       int[] frequency = new int[5];
// a, e, i, o, u

          // Count the frequency of
each vowel
         for (int i = 0; i < input.
length(); i++) {
          char ch = input.charAt(i);
            switch (ch) {
                case 'a':
                    frequency[0]++;
                    break;
                case 'e':
                    frequency[1]++;
                    break;
                case 'i':
                    frequency[2]++;
                    break;
                case 'o':
                    frequency[3]++;
                    break;
                case 'u':
                    frequency[4]++;
                    break;
            }
        }

        // Display the results
           System.out.println("Vowel
Frequency");
       System.out.println("a       "
+ frequency[0]);
       System.out.println("e       "
+ frequency[1]);
       System.out.println("i       "
+ frequency[2]);
       System.out.println("o       "
+ frequency[3]);
       System.out.println("u       "
+ frequency[4]);
    }
}
```

81. Write a program to input a string in uppercase and print the frequency of each character.
Example:
INPUT: COMPUTER HARDWARE
OUTPUT:

Characters	Frequency
A	2
C	1
D	1
E	2
H	1
M	1
O	1
P	1
R	3
T	1
U	1
W	1

Ans : COMP 2006

```
import java.util.*;
public class frequency {
     public static void main(String
s[]) {
         System.out.println("Enter a
string");
      Scanner sc = new Scanner(System.
in);
        String str = sc.nextLine();
```

NODIA

Ans : MAIN 2011

```
import java.util.*;
public class Q8 {
     public static void main(String
args[]) {
      Scanner sc = new Scanner(System.
in);
          System.out.println("Please
enter a string:");
        String a = sc.next();
int len = a.length();
        String r = "";
         for (int i = len 1; i >= 0;
i--) {
            r = r + a.charAt(i);
        }
if(t.compareTo(a) == 0)
        System.out.print("PALINDROME
STRING");
else
   System.out.print("NOT A PALINDROME
STRING");
}
}
```

86. Consider the following statement:
"January 26 is celebrated as the Republic Day of India."
Write a program to change 26 to 15, January to August, Republic to Independence, and finally print:
"August 15 is celebrated as the Independence Day of India."

Ans : SQP 2007

```
public class Republic {
     public static void main(String
args[]) {
         String str = "JANUARY 26 IS
CELEBRATED AS REPUBLIC DAY OF INDIA";
      int pos1 = str.indexOf("JANUARY
26");
        int l = str.length();
            int l3 = "JANUARY 26".
length();
       String str2 = str.substring(l3
+ 1, l);
          String s = "AUGUST 15 " +
str2;
        int len1 = s.length();
     int pos3 = s.indexOf("REPUBLIC");
       int l2 = "REPUBLIC".length();
        String str3 = s.substring(0,
pos3);
       String str4 = s.substring(pos3
+ l2 + 1, len1);
          System.out.print("The final
string is: " + str3 + "INDEPENDENCE"
+ str4);
    }
}
```

87. Write a complete Java program to take two strings str1 and str2 as input from the user. Reverse both strings individually and concatenate the reversed strings. Display the final concatenated result.

Ans : COMP 2008

```
import java.util.Scanner;
public class ReverseAndConcatenate {
    // Method to reverse a string
          public   static   String
reverseString(String str) {
        StringBuilder reversed = new
StringBuilder(str);
         return reversed.reverse().
toString();
    }
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
        // Input the first string
         System.out.print("Enter the
first string (str1): ");
        String str1 = sc.nextLine();
        // Input the second string
         System.out.print("Enter the
second string (str2): ");
        String str2 = sc.nextLine();
        // Reverse both strings
             String  reversedStr1  =
reverseString(str1);
             String  reversedStr2  =
reverseString(str2);
         // Concatenate the reversed
strings
        String result = reversedStr1
+ reversedStr2;
     // Display the final concatenated
result
          System.out.println("Final
concatenated result: " + result);
    }
}
```

NODIA

Download INDIA's Best Study app NODIA

```
x=len-1;
len=x41;
}
                String str1 = nam.
substring(0, x);
                String str2 = nam.
substring(x + len, l);
                String str3 = str1.
concat(n1);
           str3 = str3.concat(str2);

               System.out.print("The
resulting string is: " + str3);
        } else {
              System.out.print("Name
not found to be replaced");
        }
    }

     public static void main(String
args[]) {
        computeGrade(A);
            replaceName("Shiva Kumar
Sardesai");
    }
}
```

94. Write a program using a method void counter(String subject) that counts the number of digits, lowercase alphabets, uppercase alphabets, and special characters and finally displays them.

Ans : MAIN 2003

```
class Processing {
   public static void counter(String
subject) {
        int dig, up, lw, sp;
         dig = 0; up = 0; lw = 0; sp
= 0;

        int len = subject.length();
        char c;

         for (int i = 0; i <= len 1;
i++) {
              c = subject.charAt(i);
// Extracting character
           int d = c;              //
Getting ASCII code

             if (d >= 48 && d <= 57)
// ASCII range for digits 0 to 9
                dig = dig + 1;
            else if (d >= 65 && d <=
90)  // ASCII range for uppercase A-Z
                up = up + 1;
            else if (d >= 97 && d <=
122)   // ASCII range for lowercase
a-z
                lw = lw + 1;
            else
                sp = sp + 1;
        }

          System.out.println("No. of
digits: " + dig);
          System.out.println("No. of
uppercase alphabets: " + up);
          System.out.println("No. of
lowercase alphabets: " + lw);
          System.out.println("No. of
special characters: " + sp);
    }
}
```

95. Define a method stringReverse(String object) that returns the reversed string object to the main module, where it is displayed.

Ans : SQP 2002

```
public class VowelCounter {
      // Define the method to count
vowels
  public static int countVowels(String
object) {
         int count = 0;
        String vowels = "aeiouAEIOU";
       // Loop through each character
in the string
         for (int i = 0; i < object.
length(); i++) {
          char ch = object.charAt(i);
            // Check if the character
is a vowel
              if (vowels.indexOf(ch)
!= -1) {
                  count++;
             }
        }
         return count; // Return the
count to the main method
    }
    public static void main(String[]
args) {
        // Example input
         String input = "Programming
is fun";
       // Call the countVowels method
```

Downloaded from www.pdf.tube

```
import java.util.*;
public class {
    static void replaceString(String
h) {
        int x = h.indexOf("an");
        if (x >= 0) {
            int len = "an".length();
            int l = h.length();
            if (x == 0) {
                x = len 1;
                len = x + 1;
            }
         String str1 = h.substring(0,
x len + 1);
          String str2 = h.substring(x
+ len, l);
                String str3 = str1.
concat(" ");
           str3 = str3.concat(str2);
               System.out.print("The
resulting string is: " + str3);
        } else {
            System.out.print("an not
found to be replaced");
        }
    }
public  static  void  main(String
args[]) {
     Scanner inp = new Scanner(System.
in);
            System.out.print("Please
enter a string: ");
        String sh = inp.next();
        replaceString(sh);
    }
}
```

92. (i) Define a method that finds the number of digits in a number entered by the user from the keyboard. Show the use of string functions.

(ii) Define another method in the class that replaces the first digit with 0.

Ans : SQP 2019

```
public class  {
    static void find(String str) {
        int len = str.length();
        System.out.print("The number
of digits in the number = " + len);
    }

    static void repl(String a) {
        char x = a.charAt(0);
        String j = a.replace(x, 0);
          System.out.print("The final
string: " + j);
    }

     public static void main(String
args[]) {
        String f = "2567";
        find(f);
        repl(f);
    }
}
```

93. Define a class Student that has the following member methods and data members:

Data members: roll_num, name, marksEng, marksMaths, marksComp, total, percent, and grade.

Member methods:

(i) computeGrade() to assign a grade to the variable grade.

(ii) replaceName() to replace a word in the name of a student with the desired word as entered by the user.

Ans : COMP 2009

```
import java.util.*;
public class {
     static int roll_num, marksEng,
marksMaths, marksComp;
    static double total, percent;
    static char grade;
    static String name;

    static void computeGrade(char g)
{  // Method to assign grade
        grade = g;
    }

     static void replaceName(String
nam) {  // Method to replace the name
     Scanner sc = new Scanner(System.
in);
            System.out.print("Please
enter the word to replace: ");
        String n = sc.next();
            System.out.print("Please
enter the new word: ");
        String n1 = sc.next();
        int x = nam.indexOf(n);

        if (x >= 0) {
            int len = n.length();
            int l = nam.length();
```

NODIA

BETTER THAN BOOK IN THE MARKET

Why students are choosing this book 👉

School teachers already use this content to set papers.

Still not sure?

- 👉 Match this PDF with your question papers
- 👉 See the similarity yourself

Why does this happen?

- ✓ Teachers have its PDF & Word files
- ✓ Easy access
- ✓ Direct copy-paste questions
- ✓ Same pattern followed

Don't miss out—get the advantage today!

```
args[]) {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the string :");
        String nam = buf.nextLine();
              StringBuffer t = new
StringBuffer(nam);
          System.out.println("Please
enter the position from where to
extract:");
        int pos = buf.nextInt();
          System.out.println("Please
enter the number of characters to
extract:");
        int nm = buf.nextInt();
          StringBuffer strr = rev(t,
pos, nm);
        System.out.print(strr);
    }
}
```

99. Define a class Increment that contains the following members:
Data members: count, age, name, code
Constructor: Initializes count to zero value.
Methods :
(i) count Increment() to increase the value of count by 1.
(ii) extract-digits() to extract last four digits of code.
(iii) check_age() to increment the count when age is less than 18 years. At the same time, prints the last four extracted digits of the code.
(iv) Main method to call check_age().

Ans : COMP 2021

```
import java.util.*;
public class Increment {
    int count, age;
    String name, code;

    Increment() {
        count = 0;
    }

    void countIncrement() {
        count++;
    }

    String extractDigits() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the code: ");
        code = buf.next();
        int len = code.length();
            String lastFour = code.
substring(len 4, len);
        return lastFour;
    }

    void checkAge() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the age: ");
        age = buf.nextInt();
        if (age < 18) {
            countIncrement();
          String l = extractDigits();
              System.out.println("The
extracted last four digits of code
are: " + l);
        }
    }
     public static void main(String
args[]) {
      Increment obj = new Increment();
        obj.checkAge();
    }
}
```

100. Write a java method to read a string from console and replace all occurrences of the sixth character of the string by the digit 9.
/* Hint : use charAt() and replace() */

Ans : MAIN 2024

```
import java.util.*;
public class Q18 {
    static void replacing() {
     Scanner buf = new Scanner(System.
in);
          System.out.println("Please
enter the string:");
        String nam = buf.next();
        char x = nam.charAt(5);
       String ch = nam.replace(x, 9);
            System.out.print("Changed
string is :" + ch);
    }

     public static void main(String
args[]) {
        replacing();
    }
}
```

Click Here to Buy book on Amazon

```
                    int vowelCount =
countVowels(input);
        // Display the result
        System.out.println("Original
String: " + input);
       System.out.println("Number of
vowels: " + vowelCount);
    }
}
```

96. Define a method stringReverse(String object) that returns the reversed string object to the main module, where it is displayed.

Ans : COMP 2001

```
public class {
                static    StringBuffer
stringReverse(StringBuffer str) {
        str.reverse();
        return str;
    }

     public static void main(String
args[]) {
                StringBuffer t = new
StringBuffer("Aditi");
                 StringBuffer strr =
stringReverse(t);
        System.out.print(strr);
    }
}
```

97. Define a method replace(String str, int m, int n, char c) where m denotes the starting position in the string, and n denotes the number of characters to be replaced starting from position m. The method replaces the specified n characters with the character c and returns the modified string to the main method, where it is displayed.

Ans : MAIN 2005

```
public class ReplaceCharacters {
    // Define the method
  public static String replace(String
str, int m, int n, char c) {
        // Ensure valid input
       if (m < 0 || m >= str.length()
|| m + n > str.length()) {
                return "Invalid range
specified";
        }
        // Create the modified string
         StringBuilder result = new
StringBuilder(str);
     // Replace n characters starting
from position m with character c
         for (int i = m; i < m + n;
i++) {
            result.setCharAt(i, c);
        }

        return result.toString();
    }

    public static void main(String[]
args) {
        // Example usage
       String input = "programming";
      int m = 3; // Starting position
          int n = 4; // Number of
characters to replace
        char c = '*'; // Replacement
character

        // Call the replace method
       String result = replace(input,
m, n, c);

        // Display the result
        System.out.println("Original
String: " + input);
         System.out.println("Modified
String: " + result);
    }
}
```

98. Define a method rev(String, m, n) where m denotes the position in string and n denotes the number of character from m position. The method reverses the n characters specified from position m and returns the reversed string to main method where it is displayed.

Ans : SQP 2017

```
import java.util.*;
public class {
   static StringBuffer rev(StringBuffer
str, int m, int n) {
        /* Note: here return type is
StringBuffer */
       String strn = str.substring(m,
m + n);
             StringBuffer str2 = new
StringBuffer(strn);
        str2.reverse();
        return str2;
    }
     public static void main(String
```

NODIA

```
        for (i = 0; i <= l 1; i++) {
             char x = str.charAt(i);
// Extract character
             int a = x;  // Get the
ASCII code

if (a >= 65 && a <= 90) {
   char c = Character.toLowerCase(x);
    s = s + c;
}
else if (a >= 97 && a <= 120) {
   char d = Character.toUpperCase(x);
    s = s + d;
}
else {
    s = s + x;
}
System.out.print(s);
}
```

Variables Used :
str: Inputted string.
l: Stores length of the string.
x: Stores character extracted at index i.
s: Stores the new string.

84. Design a class to overload a function Joystring() as follows:

(i) void Joystring(String s, char ch1, char ch2) This method takes one string argument and two character arguments, replaces the character argument ch1 with ch2 in the given string s, and prints the new string.
Example:
Input: s = "TECHNALAGY", ch1 = A, ch2 = O
Output: "TECHNOLOGY"

(ii) void Joystring(String s) This method takes one string argument, prints the position of the first space, and the last space of the given string s.
Example:
Input: "Cloud computing means Internet based computing"
Output:
First index: 5
Last index: 36

(iii) void Joystring(String s1, String s2) This method takes two string arguments, combines the two strings with a space between them, and prints the resultant string.
Example:
Input: s1 = "COMMON WEALTH", s2 = "GAMES"
Output: "COMMON WEALTH GAMES"

Ans : COMP 2014

```
public class JoyStringClass {
    static void joyString(String s,
char ch1, char ch2) {
        String str = "";
      // Implementation continues...
int indx = s.indexOf(ch1);

if (indx > -1) {
    str = s.replace(ch1, ch2);
} else {
    System.out.println("Char " + ch1
+ " does not exist in " + s);
          System.out.println("Output
String:");
    System.out.println(str);
}

static void joyString(String s) {
   int firstIndexOfSpace = s.indexOf(
);
          int lastIndexOfSpace =
s.lastIndexOf( );
    System.out.println("First Index
of Space = " + firstIndexOfSpace);
   System.out.println("Last Index of
Space = " + lastIndexOfSpace);
}

static void joyString(String s1,
String s2) {
         String s3 = s1.concat("
").concat(s2);
    System.out.println(s3);
}
public static void main(String[]
args) {
    joyString("Technalagy", a, o);
    joyString("Cloud is rain");
    joyString("Cloud", "rain");
}
```

Variables Used :
s1, s2 : Strings to operate upon.
ch1, ch2 : Characters to replace.
indx : To store the index or position.

85. Write a program using a method Palin() to check whether a string is a palindrome or not. A palindrome is a string that reads the same from left to right and vice versa.
Example: MADAM, ARORA, ABBA, etc.

Downloaded from www.pdf.tube

101. Write a Java method swapCase(String input1, String input2) that converts the first string (input1) to uppercase and the second string (input2) to lowercase. Display the converted results.

Ans : SQP 2023

```
public class SwapCaseExample {
   public static void swapCase(String input1, String input2) {
        // Convert the first string to uppercase
        String upperCaseStr = input1.toUpperCase();
        // Convert the second string to lowercase
        String lowerCaseStr = input2.toLowerCase();
        // Display the results
     System.out.println("First string in uppercase: " + upperCaseStr);
    System.out.println("Second string in lowercase: " + lowerCaseStr);
    }
    public static void main(String[] args) {
        // Example usage
        swapCase("hello World", "JAVA Programming");
    }
}
```

102. Write a program that accepts a time in the format hh:mm:ss and separates the hours, minutes, and seconds as integer numbers. Display the extracted values.

Ans : COMP 2017

```
import java.util.Scanner;

public class TimeSeparator {
    public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);

        // Accept the time input
        System.out.print("Enter time in hh:mm:ss format: ");
        String time = sc.nextLine();

       // Split the time into hours, minutes, and seconds
         String[] timeParts = time.split(":");
             int hours = Integer.parseInt(timeParts[0]);
            int minutes = Integer.parseInt(timeParts[1]);
            int seconds = Integer.parseInt(timeParts[2]);

          // Display the extracted values
        System.out.println("Hours: " + hours);
        System.out.println("Minutes: " + minutes);
        System.out.println("Seconds: " + seconds);
    }
}
```

103. Write a program that accepts a date in the format dd/mm/yyyy, and separates the day, the month and the year as integer numbers.

Ans : MAIN 2016

```
import java.util.*;
public class {
     public static void main(String args[]) {
Scanner buf = new Scanner(System.in);
System.out.print("Please enter a string:");
String date = buf.next();
char a = date.charAt(0);
char b = date.charAt(1);
int d1 = a 48;
int d2 = b 48;
char c = date.charAt(3);
int m1 = c 48;
char d = date.charAt(4);
int m2 = d 48;
char e = date.charAt(6);
int y1 = e 48;
char f = date.charAt(7);
int y2 = f 48;
char g = date.charAt(8);
int y3 = g 48;
char h = date.charAt(9);
int y4 = h 48;
System.out.println();
System.out.print("DATE" + d1 + d2 + "-" + m1 + m2 + "-" + y1 + y2 + y3 + y4);
}
}
```

NODIA

104. Consider that the age of a person is read as "10 years old in the year 1960", write a java method that changes the age according if the current year is 2005.

Ans : SQP 2020

```
public class {
     public static void main(String
args[]) {
        String str = "10 years old in
the year 1960";
         int l = str.length();
         int l2 = "1960".length();
         int st = l l2;
         char buf[] = new char[10];
         str.getChars(st, l, buf, 0);
        int yr = (buf[0] 48) * 1000 +
(buf[1] 48) * 100 + (buf[2] 48) * 10
+ (buf[3] 48) * 1;
         int diff = 2000 yr;
         char a[] = new char[10];
         str.getChars(0, 2, a, 0);
          int age = (a[0] 48) * 10 +
(a[1] 48) * 1;
         int f = age + diff;
       String n = f + str.substring(2,
l 4) + 2005;
       System.out.println("The string
is finally changed: " + n);
    }
}
```

105. Write a java method convertCase(String obj1, String obj2) that converts obj1 to lowercase and obj2 to uppercase. Display the results.

Ans : COMP 2004

```
import java.util.*;
public class {
     static void convertCase(String
obj1, String obj2) {

     String str1 = obj1.toLowerCase();
     String str2 = obj2.toUpperCase();
       System.out.println("The string
" + obj1 + " in lower case is : " +
str1);

       System.out.println("The string
" + obj2 + " in upper case is : " +
str2);
    }

     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
         System.out.print("Enter two
strings please: ");
        String s1 = buf.next();
        String s2 = buf.next();
        convertCase(s1, s2);
    }
}
```

106. Write a java method that accepts two strings as parameters, compares them exactly and returns true if they match and false if they dont match.

Ans : MAIN 2006

```
import java.util.*;
public class {
    static boolean comparison(String
str1, String str2) {
       if (str1.equals(str2)) return
true;
        else return false;
    }

     public static void main(String
args[]) {
     Scanner buf = new Scanner(System.
in);
         System.out.print("Enter two
strings please: ");
        String s1 = buf.next();
        String s2 = buf.next();
         boolean v = comparison(s1,
s2);
         if (v == true) System.out.
print("The strings are same ");
          else System.out.print("The
strings are not same ");
    }
}
```

107. Write a Java method that extracts the last three characters of one string and the first three characters of another string. It compares these two extracted parts and prints "MATCH" if they are the same, otherwise, it prints "NO MATCH".

Ans : SQP 2010

```
import java.util.Scanner;

public class StringComparison {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
```

Click Here to Buy book on Amazon

```
in);

        // Input two strings
         System.out.print("Enter the
first string: ");
        String str1 = sc.nextLine();

         System.out.print("Enter the
second string: ");
        String str2 = sc.nextLine();

         // Check if the strings are
long enough
       if (str1.length() < 3 || str2.
length() < 3) {
            System.out.println("Both
strings must have at least 3
characters!");
            return;
        }

          // Extract the last three
characters of the first string
             String part1 = str1.
substring(str1.length() - 3);

          // Extract the first three
characters of the second string
     String part2 = str2.substring(0,
3);

       // Compare the extracted parts
and print the result
        if (part1.equals(part2)) {
        System.out.println("MATCH");
        } else {
              System.out.println("NO
MATCH");
        }
    }
}
```

108. Write a Java method that extracts the first five characters from one string and the middle characters from another string. Compare the extracted parts and print "EQUAL" if they are the same; otherwise, print "NOT EQUAL."

Ans : MAIN 2013

```
import java.util.*;
public class {
    static void extract(String str1,
String str2) {
        int l = str2.length();
        int m = l / 2 1;
int st = (0 + m) / 2 + 1;
int end = (m + l) / 2 + 1;
String s1 = str1.substring(0, 5);
String s2 = str2.substring(st, end);
System.out.println("First        five
characters of " + str1 + " are " +
s1);
System.out.println("Mid          five
characters of " + str2 + " are " +
s2);
if    (s1.equals(s2))    System.out.
print("They are equal");
else System.out.print("They are not
equal");
}

public   static   void   main(String
args[]) {
    Scanner inp = new Scanner(System.
in);
    System.out.println("Please enter
two strings :");
    String a = inp.nextLine();
    String a2 = inp.nextLine();
    extract(a, a2);
}
}
```

109. Write a program that encodes a word into Piglatin. To translate word into a Piglatin word, convert the word into uppercase and then place the first vowel of the original word as the start of the new word along with the remaining alphabets. The alphabets present before the vowel being shifted towards the end followed by "AY".
Sample input (1): London,
Sample output (1): ONDONLAY
Sample input (2): Olympics,
Sample output (2): OLYMPICSAY

Ans : SQP 2014

```
import java.util.Scanner;
public class Piglatin {
   public static void main(String[]
args) {
    Scanner scan = new Scanner(System.
in);
         System.out.println("Enter a
string:");
          String str = scan.next().
toUpperCase();
        char ch;
         for (int i = 0; i <= str.
```

Downloaded from www.pdf.tube

```
length() 1; i++) {
            ch = str.charAt(i);
           if (ch == A || ch == E ||
ch == I || ch == O || ch == U) {
                    String s1 = str.
substring(0, i);
                    String s2 = str.
substring(i);
                        String s3 =
s2.concat(s1).concat("AY");
             System.out.println(s3);
                break;
            }
        }
    }
}
```

Variables used :
str: input String to check Piglatin
ch: character of the string
s1, s2: parts of Piglatin
s3: final Piglatin string

110. Write a Java program to input forty words into an array and sort them in descending alphabetical order using the selection sort technique. Display the sorted array.

Ans : COMP 2011

```
import java.util.*;
public class {
    public static void main(String[]
ar) {
        String str;
     Scanner sc = new Scanner(System.
in);
     String names[] = new String[40];
      System.out.println("enter all
the names");
       for (int i = 0; i < 40; i++) {
           names[i] = sc.nextLine();
        }

       for (int i = 0; i < 40; i++) {
            for (int j = i + 1; j <
40; j++) {
                    if (names[i].
compareTo(names[j]) < 0) {
                    str = names[i];
                names[i] = names[j];
                    names[j] = str;
                }
            }
        }
        System.out.println("names in
descending order");
       for (int i = 0; i < 40; i++) {
         System.out.println(names[i]
+ ",");
        }
    }
}
```

111. Write a Java program to input a list of twenty numbers into an array. Arrange these numbers in ascending order using the selection sort technique. Display the sorted array.

Ans : MAIN 2007

```
import java.util.Scanner;

public class SelectionSortNumbers {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Create an array to store
20 numbers
       int[ ] numbers = new int[20];

         // Input numbers into the
array
        System.out.println("Enter 20
numbers:");
       for (int i = 0; i < 20; i++) {
          numbers[i] = sc.nextInt();
        }

        // Selection sort algorithm
to sort in ascending order
        for (int i = 0; i < numbers.
length - 1; i++) {
            int minIndex = i;
             for (int j = i + 1; j <
numbers.length; j++) {
                  if (numbers[j] <
numbers[minIndex]) {
                   minIndex = j;
               }
            }
            // Swap the elements
        int temp = numbers[minIndex];
                numbers[minIndex] =
numbers[i];
            numbers[i] = temp;
        }
```

NODIA

```
        // Print the sorted array
          System.out.println("Sorted
array in ascending order:");
        for (int num : numbers) {
             System.out.print(num +
" ");
        }
    }
}
```

112. Write a Java program to accept a string and reverse the order of words in it. Display the reversed string.
Sample Input: "Welcome to the programming world"
Sample Output: "world programming the to Welcome"

Ans : SQP 2008

```
import java.util.Scanner;

public class ReverseWords {
  public static void main(String[ ]
args) {
     Scanner sc = new Scanner(System.
in);

        // Accept the input string
          System.out.print("Enter a
string: ");
       String input = sc.nextLine();

       // Split the string into words
       String[] words = input.split("
");

        // Reverse the words
        StringBuilder reversedString
= new StringBuilder();
       for (int i = words.length - 1;
i >= 0; i--) {
                    reversedString.
append(words[i]).append(" ");
        }

        // Print the reversed string
        System.out.println("Reversed
String:    "    +    reversedString.
toString().trim());
    }
}
```

113. Write a Java program to take a string in lowercase and convert the first letter of each word to uppercase. Display the modified string. Sample Input: we are in cyber world Sample Output: We Are In Cyber World

Ans : COMP 2022

```
import java.util.*;
class Capitalise {
    public static void main() {
        boolean flag = false;
         System.out.println("enter a
sentence in lower case to capitalise
first letter of each word");
     Scanner sc = new Scanner(System.
in);
        String g = sc.nextLine();
String s = g.trim();
String n = "";
char  ch  =  Character.toUpperCase(s.
charAt(0));
n = n + ch;
for  (int  i  =  1;  i  <=  s.length()  2;
i++)  {
       if  ((s.charAt(i)  ==    )  &&
(s.charAt(i + 1) !=  )) {
       ch = Character.toUpperCase(s.
charAt(i + 1));
        n = n + ch;
    } else {
        ch = s.charAt(i);
        n = n + ch;
    }
}
n = n + s.charAt(s.length() 1);
System.out.println(n);
}
}
```

114. Define a class to accept a String and print the number of digits, alphabets, and special characters in the string.
Example:
S = "KAPILDEV@83"
Output:
Number of digits = 2
Number of Alphabets = 8
Number of Special characters = 1

Ans : MAIN 2015

```
import java.util.Scanner;

class Count {
     public static void main(String
```

Download INDIA's Best Study app NODIA

Click Here to Buy book on Amazon

Visit :
www.pdf.tube

Download PDFs For :

- ✓ Chapterwise Question Bank
- ✓ JEE NEET Foundation
- ✓ Primary Classes Worksheets
- ✓ Olympiad Guide
- ✓ Sample Papers
- ✓ Previous Year Papers
- ✓ Previous JEE Papers
- ✓ Previous NEET Papers

NODIA

Nodia Editions LLP
125, Sector 6, Vidhyadhar Nagar, Jaipur 302039

MRP ₹ 300.00

Ans : COMP 2019

```
import java.util.Scanner;
public class CountStringComponents {
    public static void main(String[
] args) {
     Scanner sc = new Scanner(System.
in);
        // Input the string
          System.out.print("Enter a
string: ");
       String input = sc.nextLine();
        int digits = 0, vowels = 0,
consonants = 0;
      // Loop through each character
in the string
         for (int i = 0; i < input.
length(); i++) {
              char ch = Character.
toLowerCase(input.charAt(i));

          if (Character.isDigit(ch))
{
               digits++;
               } else if ("aeiou".
indexOf(ch) != -1) {
               vowels++;
              } else if (Character.
isLetter(ch)) {
               consonants++;
            }
        }

        // Display the results
       System.out.println("Number of
digits = " + digits);
       System.out.println("Number of
vowels = " + vowels);
       System.out.println("Number of
consonants = " + consonants);
    }
}
```

117. Define a class to accept a string and print the number of special characters, uppercase letters, and lowercase letters in the string.
Example:
Input:
S = "Welcome @Home!"
Output:
Number of special characters = 2
Number of uppercase letters = 1
Number of lowercase letters = 9

Ans : MAIN 2009

```
import java.util.Scanner;

public class CountSpecialCharacters
{
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);

        // Input the string
          System.out.print("Enter a
string: ");
       String input = sc.nextLine();

      int specialChars = 0, upperCase
= 0, lowerCase = 0;

       // Loop through each character
in the string
         for (int i = 0; i < input.
length(); i++) {
          char ch = input.charAt(i);

      if (Character.isUpperCase(ch))
{
               upperCase++;
              } else if (Character.
isLowerCase(ch)) {
               lowerCase++;
             } else if (!Character.
isLetterOrDigit(ch) && !Character.
isWhitespace(ch)) {
               specialChars++;
            }
        }
        // Display the results
          System.out.println("Number
of special characters = " +
specialChars);
       System.out.println("Number of
uppercase letters = " + upperCase);
       System.out.println("Number of
lowercase letters = " + lowerCase);
    }
}
```

www.nodia.in

Downloaded from www.pdf.tube

```
args[]) {
      Scanner sc = new Scanner(System.
in);
         // Prompt the user to enter
a line of text
           System.out.print("Enter a
line of text: ");
        String text = sc.nextLine();
       // Variables to count letters,
digits, and special characters
         int letter = 0, digit = 0,
special = 0;

        // Iterate through the input
string
          for (int i = 0; i < text.
length(); i++) {
            char c = text.charAt(i);

          if (Character.isLetter(c))
{
                letter++;
              } else if (Character.
isDigit(c)) {
                digit++;
            } else {
                special++;
            }
        }

        // Print the results
       System.out.println("Number of
letters = " + letter);
       System.out.println("Number of
digits = " + digit);
       System.out.println("Number of
special characters = " + special);

        // Close the scanner
        sc.close();
    }
}
```

115. Define a class to accept a String and print the number of uppercase letters, lowercase letters, and spaces in the string.
Example:
Input:
S = "Hello World 2023"
Output:
Number of uppercase letters = 2
Number of lowercase letters = 8
Number of spaces = 2

Ans : SQP 2018

```
import java.util.Scanner;
public class CountCharacters {
    public static void main(String[]
args) {
     Scanner sc = new Scanner(System.
in);
        // Accepting the string from
the user
          System.out.print("Enter a
string: ");
        String S = sc.nextLine();
     // Variables to count uppercase,
lowercase, and spaces
        int uppercaseCount = 0;
        int lowercaseCount = 0;
        int spaceCount = 0;
        // Loop through the string
      for (int i = 0; i < S.length();
i++) {
            char ch = S.charAt(i);
       if (Character.isUpperCase(ch))
{
                uppercaseCount++;
              } else if (Character.
isLowerCase(ch)) {
                lowercaseCount++;
            } else if (ch == ' ') {
                spaceCount++;
            }
        }
        // Print the counts
          System.out.println("Number
of uppercase letters = "
uppercaseCount);
          System.out.println("Number
of lowercase letters = "
lowercaseCount);
       System.out.println("Number of
spaces = " + spaceCount);
    }
}
```

116. Define a class to accept a string and print the number of digits, vowels, and consonants in the string.
Example:
Input:
S = "Programming 101 is fun!"
Output:
Number of digits = 3
Number of vowels = 5
Number of consonants = 13

NODIA